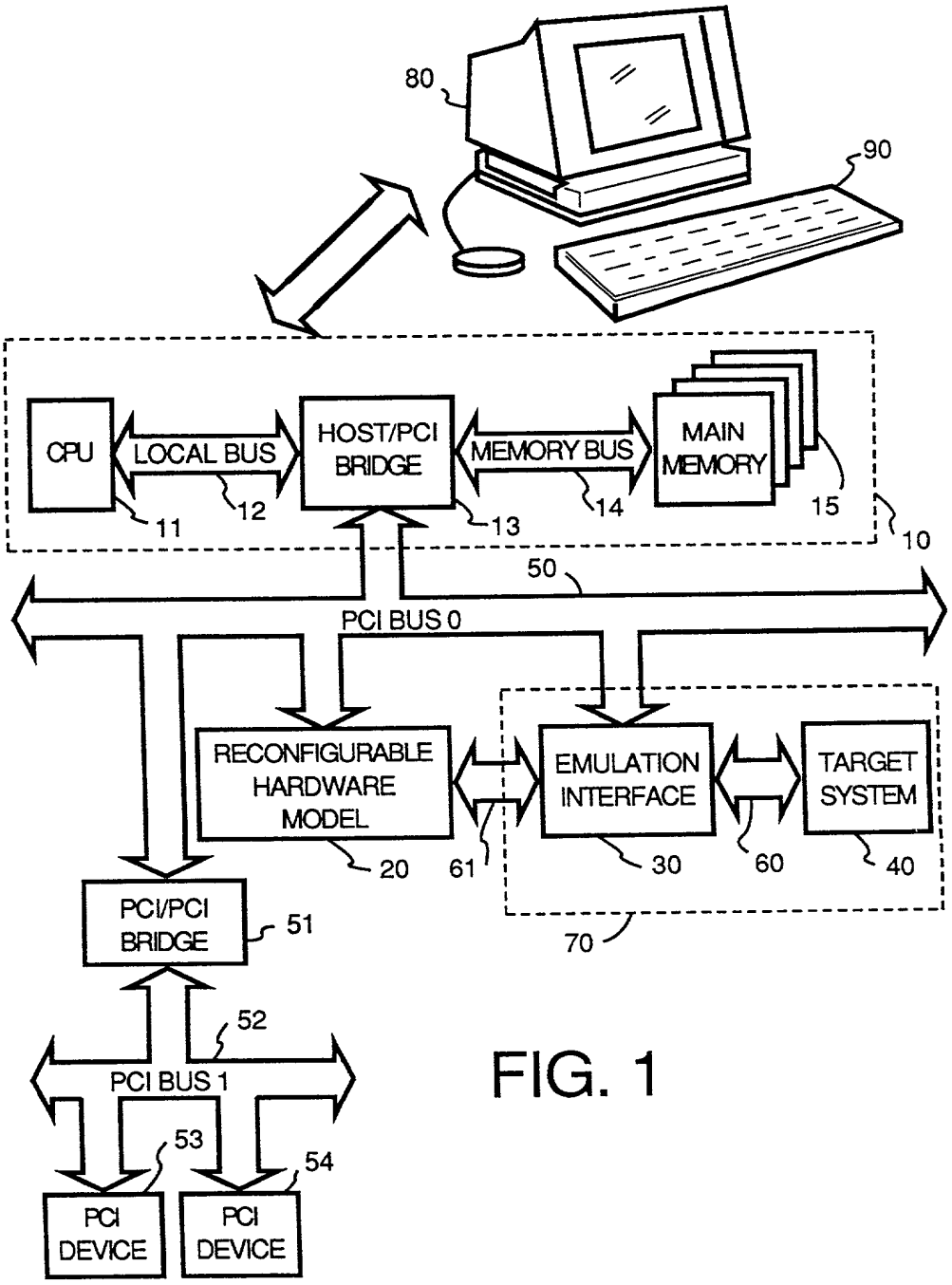

80
90
CPU
LOCAL BUS
HOST/PCI BRIDGE
MEMORY BUS
MAIN MEMORY
11
12
13
14
15
10
50
PCI BUS 0
RECONFIGURABLE HARDWARE MODEL
EMULATION INTERFACE
TARGET SYSTEM
20
61
30
60
40
70
PCI/PCI BRIDGE
51
52
PCI BUS 1
53
54
PCI DEVICE
PCI DEVICE

FIG. 1

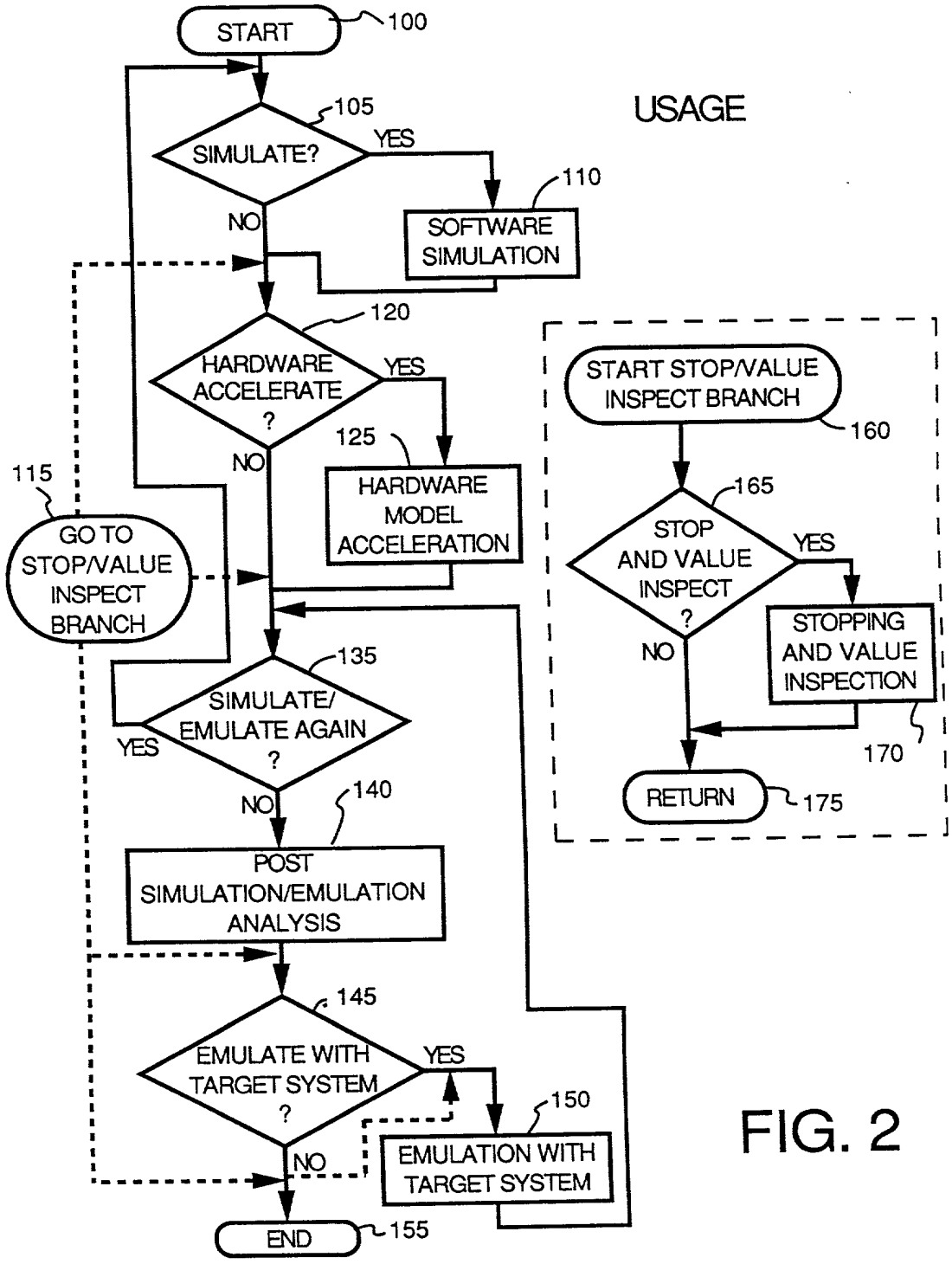

USAGE
START
100
105
SIMULATE?
YES
NO
110
SOFTWARE SIMULATION
120
HARDWARE ACCELERATE ?
YES
NO
125
HARDWARE MODEL ACCELERATION
115
GO TO STOP/VALUE INSPECT BRANCH
135
SIMULATE/ EMULATE AGAIN ?
YES
NO
140
POST SIMULATION/EMULATION ANALYSIS
145
EMULATE WITH TARGET SYSTEM ?
YES
NO
150
EMULATION WITH TARGET SYSTEM
END
155
START STOP/VALUE INSPECT BRANCH
160
165
STOP AND VALUE INSPECT ?
YES
NO
STOPPING AND VALUE INSPECTION
170
RETURN
175

FIG. 2

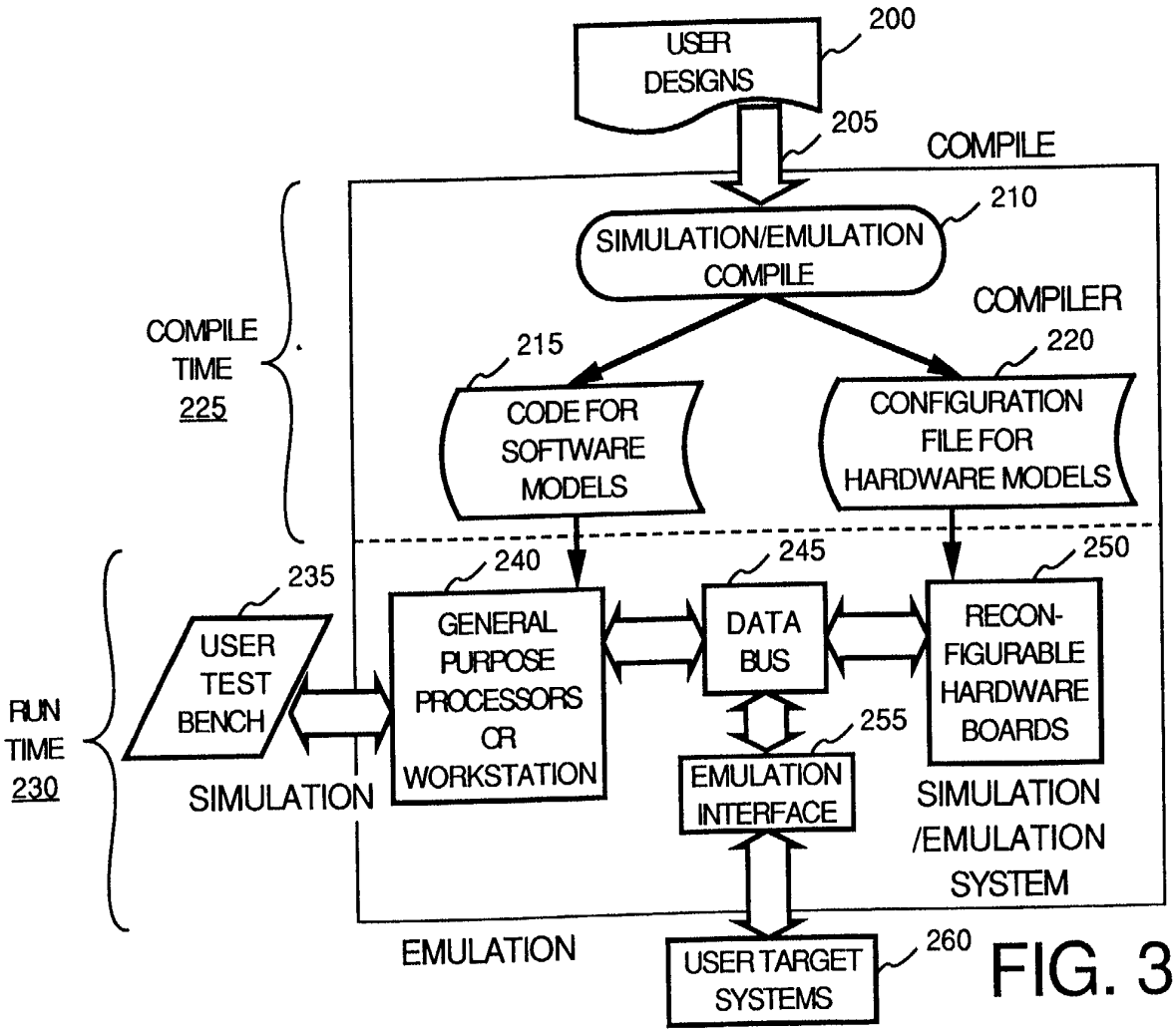


FIG. 3

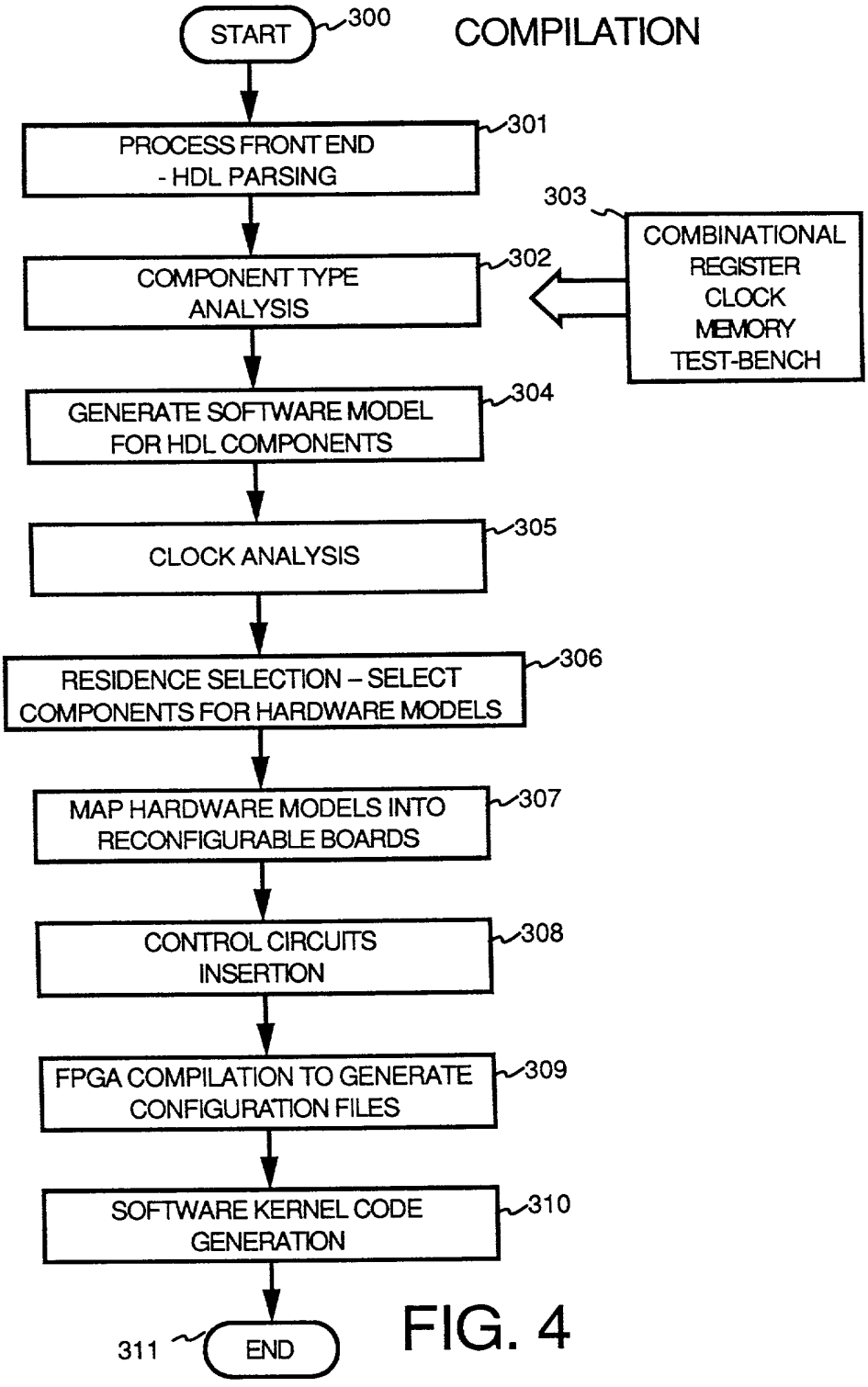

COMPILATION
START
300
PROCESS FRONT END
- HDL PARSING
301
COMPONENT TYPE
ANALYSIS
302
303
COMBINATIONAL
REGISTER
CLOCK
MEMORY
TEST-BENCH
GENERATE SOFTWARE MODEL
FOR HDL COMPONENTS
304
CLOCK ANALYSIS
305
RESIDENCE SELECTION – SELECT
COMPONENTS FOR HARDWARE MODELS
306
MAP HARDWARE MODELS INTO
RECONFIGURABLE BOARDS
307
CONTROL CIRCUITS
INSERTION
308
FPGA COMPILATION TO GENERATE
CONFIGURATION FILES
309
SOFTWARE KERNEL CODE
GENERATION
310
311
END

FIG. 4

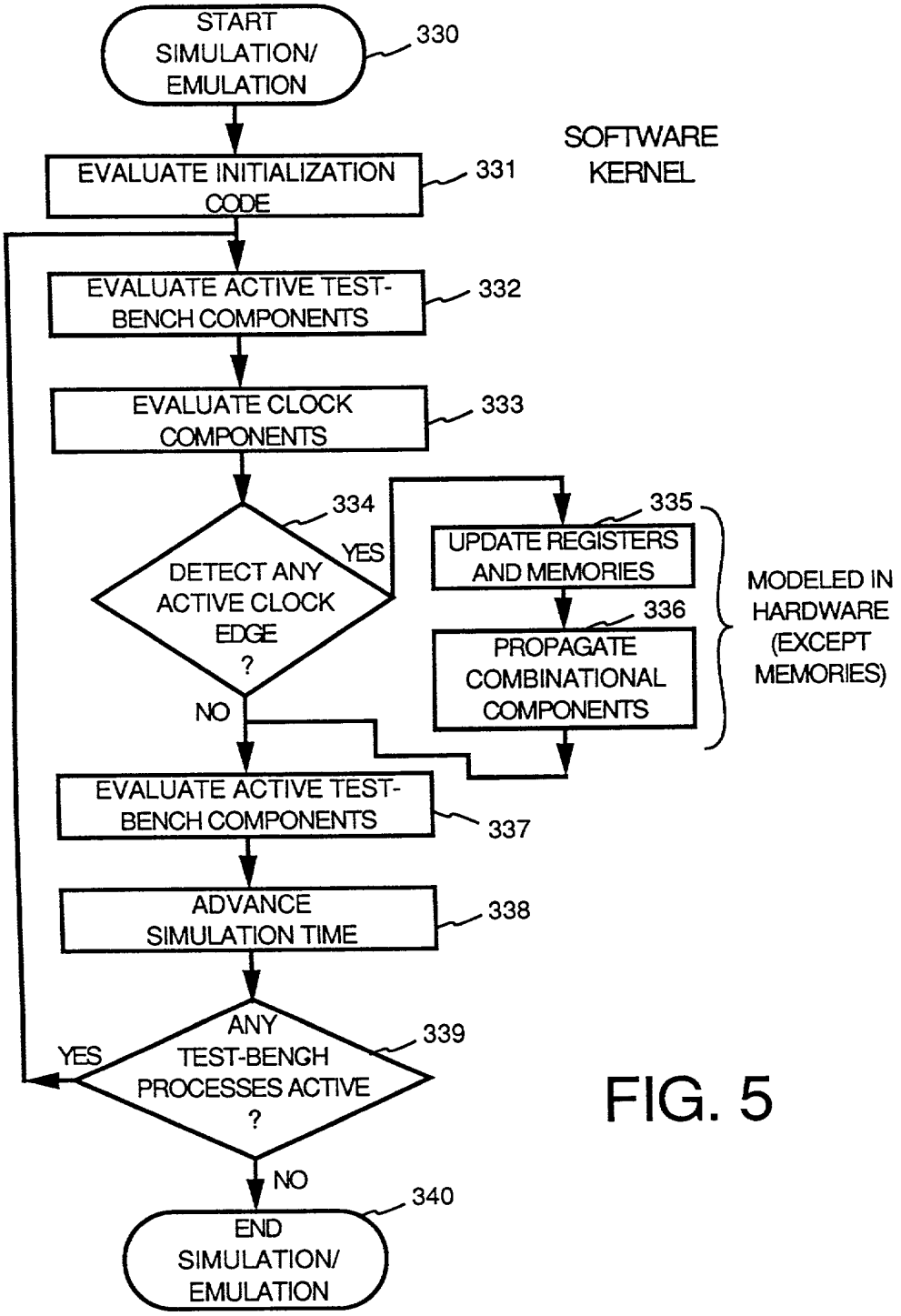

START SIMULATION/ EMULATION
330
SOFTWARE KERNEL
EVALUATE INITIALIZATION CODE
331
EVALUATE ACTIVE TEST-BENCH COMPONENTS
332
EVALUATE CLOCK COMPONENTS
333
334
DETECT ANY ACTIVE CLOCK EDGE ?
YES
NO
335
UPDATE REGISTERS AND MEMORIES
336
PROPAGATE COMBINATIONAL COMPONENTS
MODELED IN HARDWARE (EXCEPT MEMORIES)
EVALUATE ACTIVE TEST-BENCH COMPONENTS
337
ADVANCE SIMULATION TIME
338
ANY TEST-BENGH PROCESSES ACTIVE ?
339
YES
NO
340
END SIMULATION/ EMULATION

FIG. 5

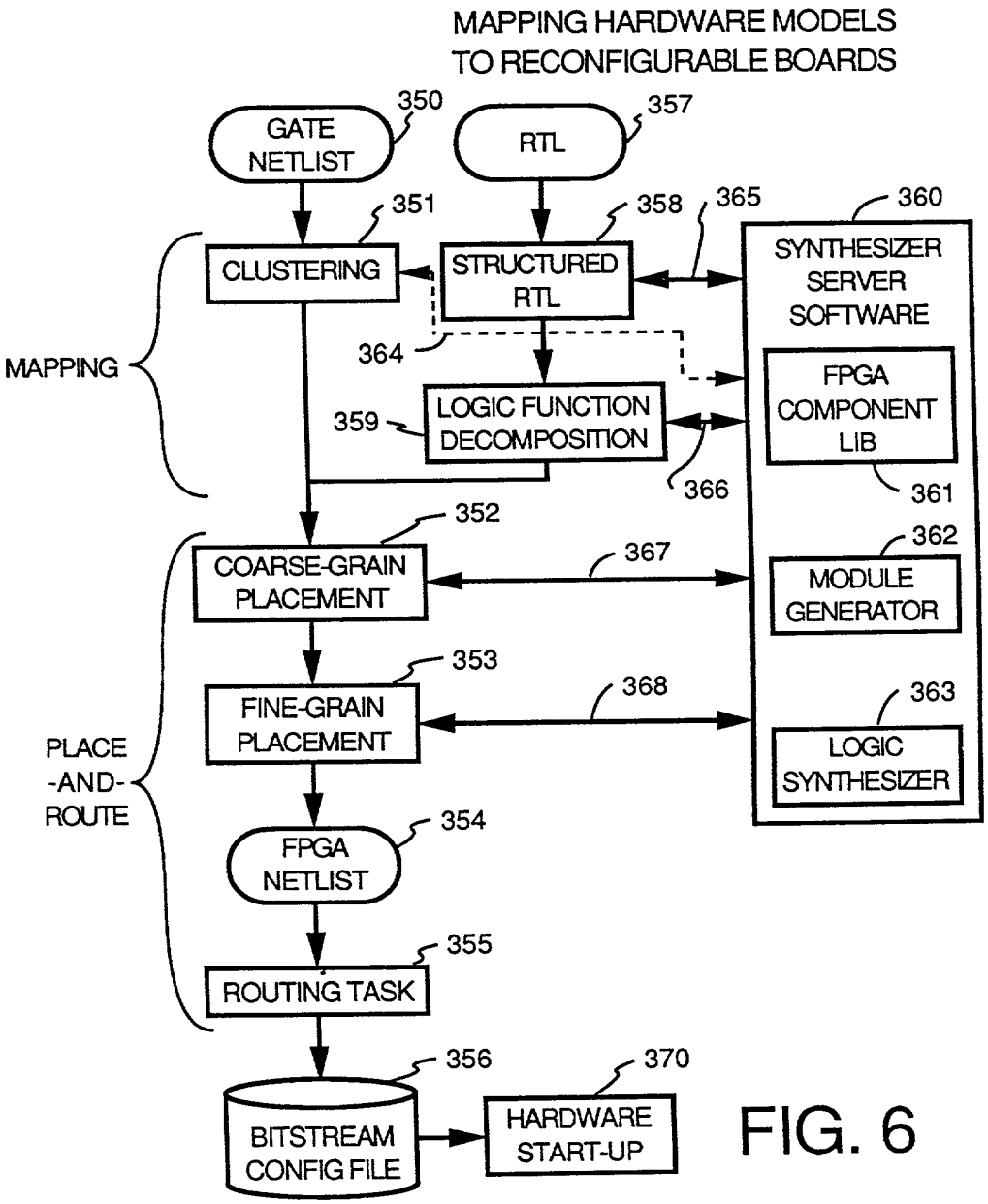

MAPPING HARDWARE MODELS
TO RECONFIGURABLE BOARDS
GATE NETLIST
350
RTL
357
CLUSTERING
351
STRUCTURED RTL
358
365
360
SYNTHESIZER SERVER SOFTWARE
MAPPING
364
FPGA COMPONENT LIB
LOGIC FUNCTION DECOMPOSITION
359
366
361
352
COARSE-GRAIN PLACEMENT
367
362
MODULE GENERATOR
353
FINE-GRAIN PLACEMENT
368
363
LOGIC SYNTHESIZER
PLACE -AND- ROUTE
FPGA NETLIST
354
ROUTING TASK
355
356
370
BITSTREAM CONFIG FILE
HARDWARE START-UP

FIG. 6

	F11	F12	F13	F14	F21	F22	F23	F24	F31	F32	F33	F34	F41	F42	F43	F44
F11	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0
F12	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0
F13	1	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0
F14	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1
F21	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0
F22	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0
F23	0	0	1	0	1	1	1	1	0	0	1	0	0	0	1	0
F24	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1
F31	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0
F32	0	1	0	0	0	1	0	0	1	1	1	1	0	1	0	0
F33	0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	0
F34	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
F41	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1
F42	0	1	0	0	0	1	0	0	0	1	0	0	1	1	1	1
F43	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	1
F44	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1

FIG. 7

FPGA INTERCONNECTION

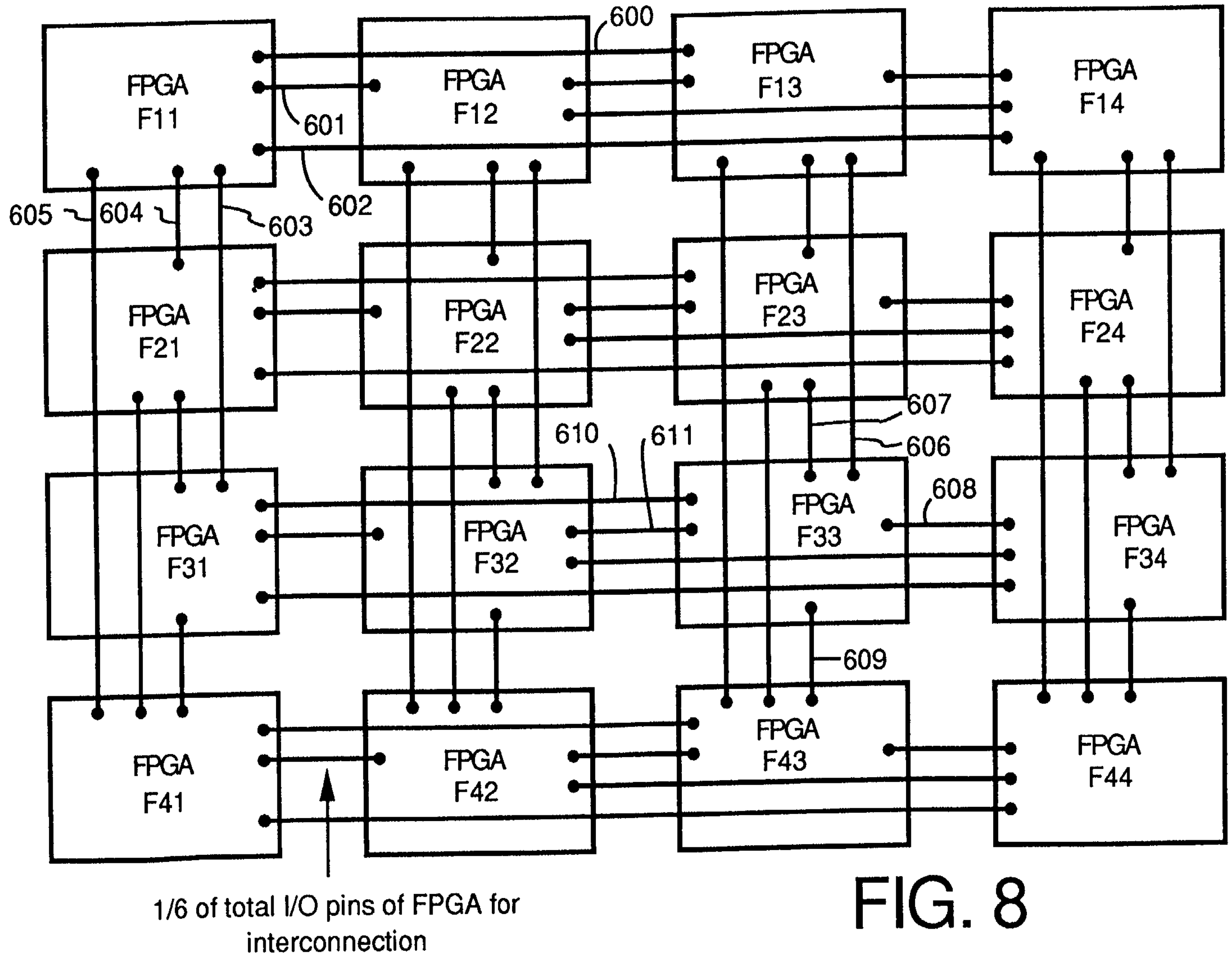


FIG. 8

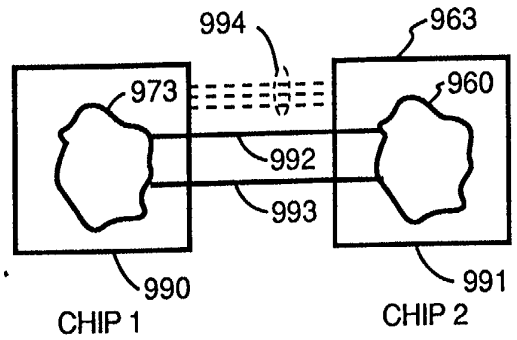


FIG. 9(a)

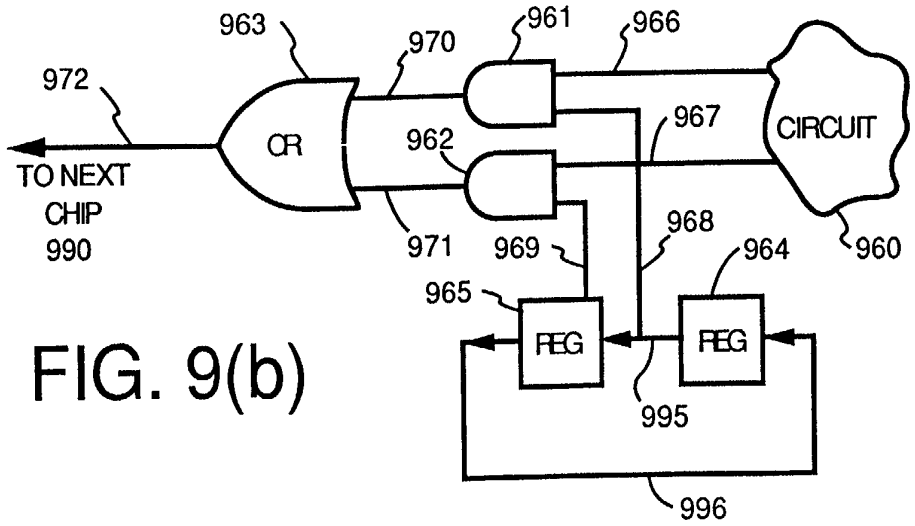


FIG. 9(b)

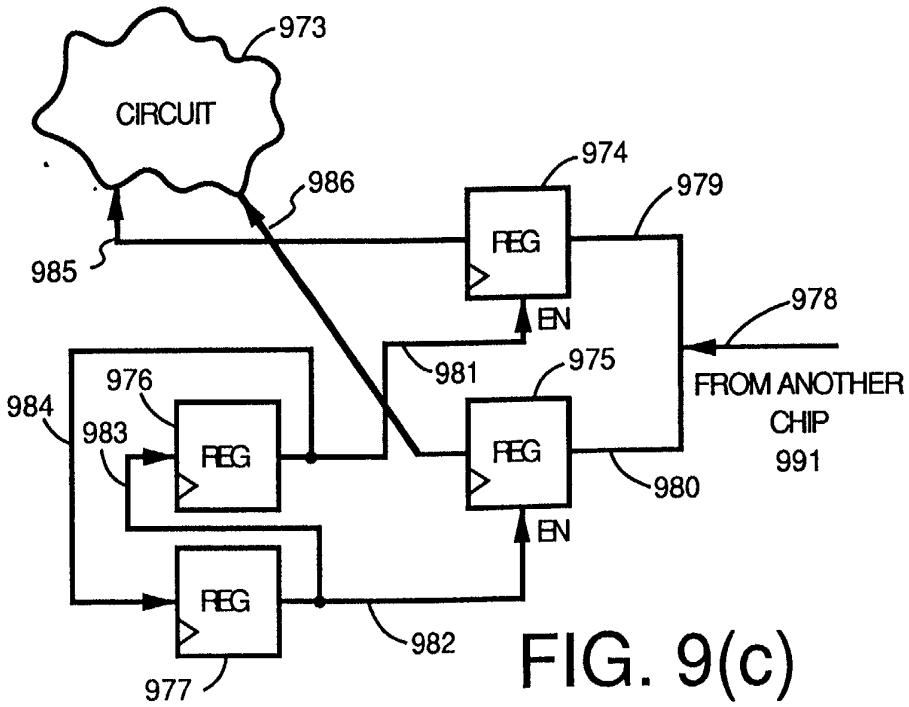


FIG. 9(c)

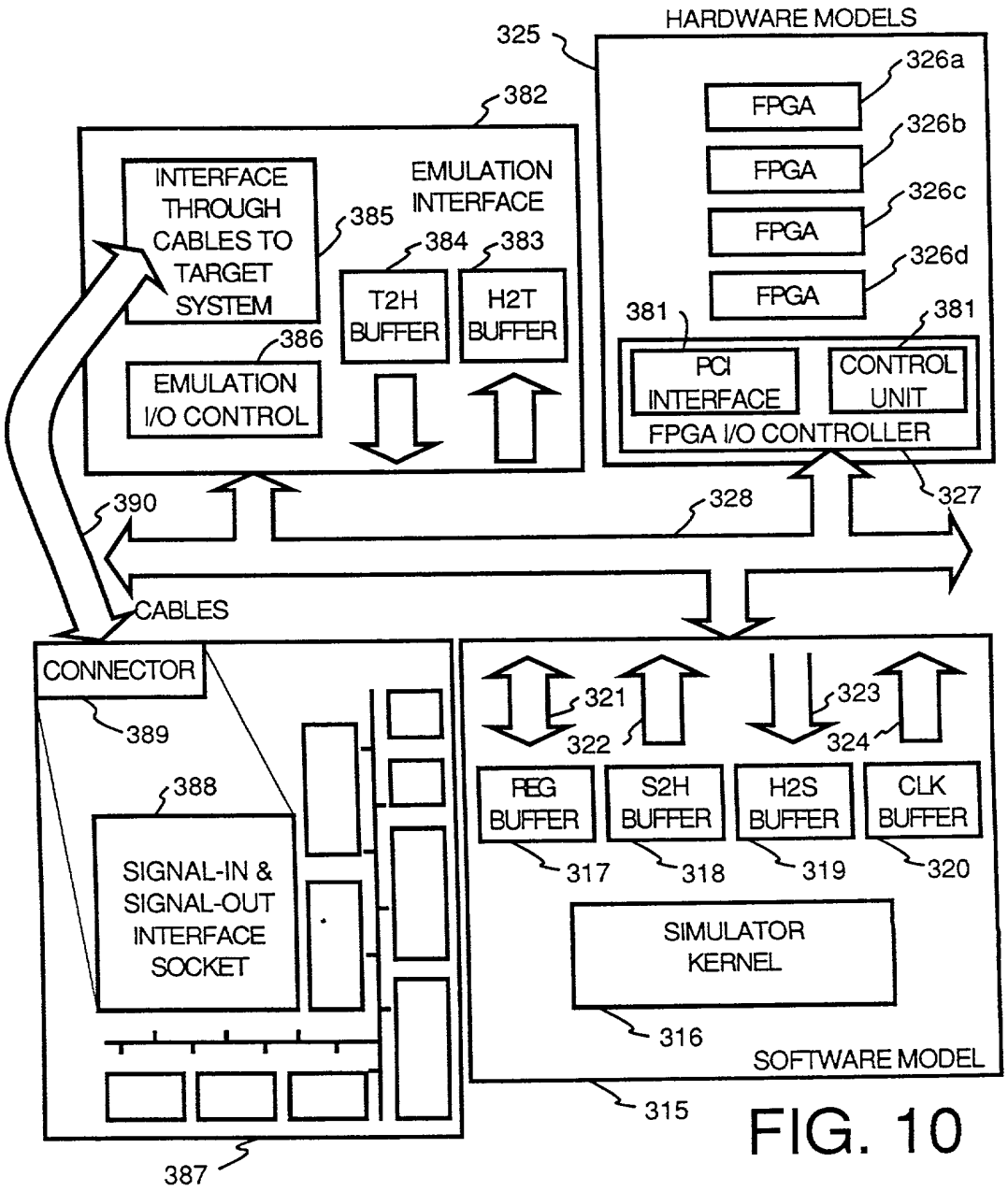

HARDWARE MODELS
325
326a
FPGA
326b
FPGA
326c
FPGA
326d
FPGA
382
INTERFACE THROUGH CABLES TO TARGET SYSTEM
385
EMULATION INTERFACE
384
383
T2H BUFFER
H2T BUFFER
386
EMULATION I/O CONTROL
381
381
PCI INTERFACE
CONTROL UNIT
FPGA I/O CONTROLLER
390
328
327
CABLES
CONNECTOR
389
388
SIGNAL-IN & SIGNAL-OUT INTERFACE SOCKET
321
322
323
324
REG BUFFER
S2H BUFFER
H2S BUFFER
CLK BUFFER
317
318
319
320
SIMULATOR KERNEL
316
SOFTWARE MODEL
315
387

FIG. 10

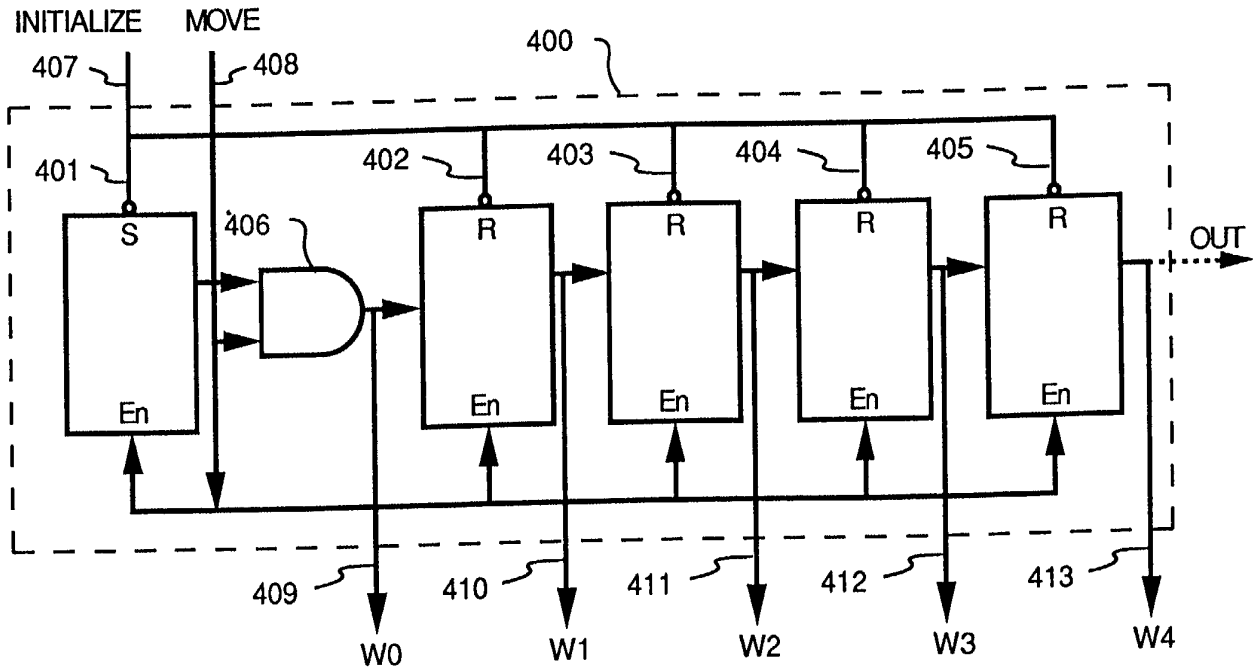


FIG. 11

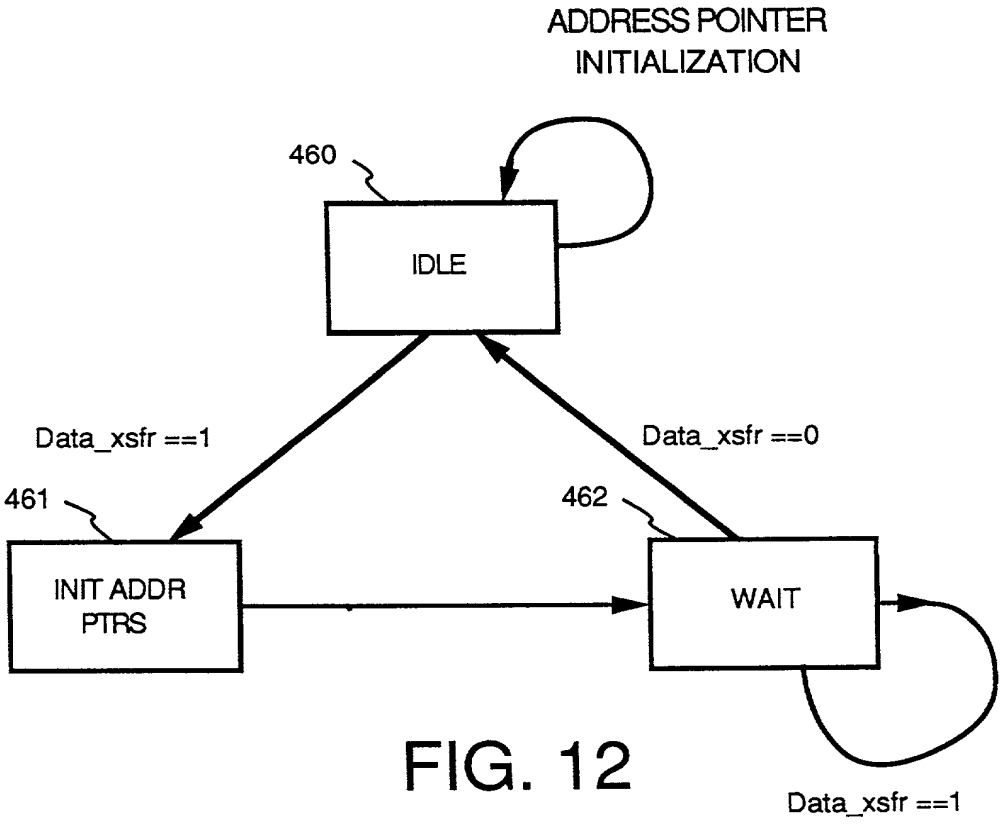


FIG. 12

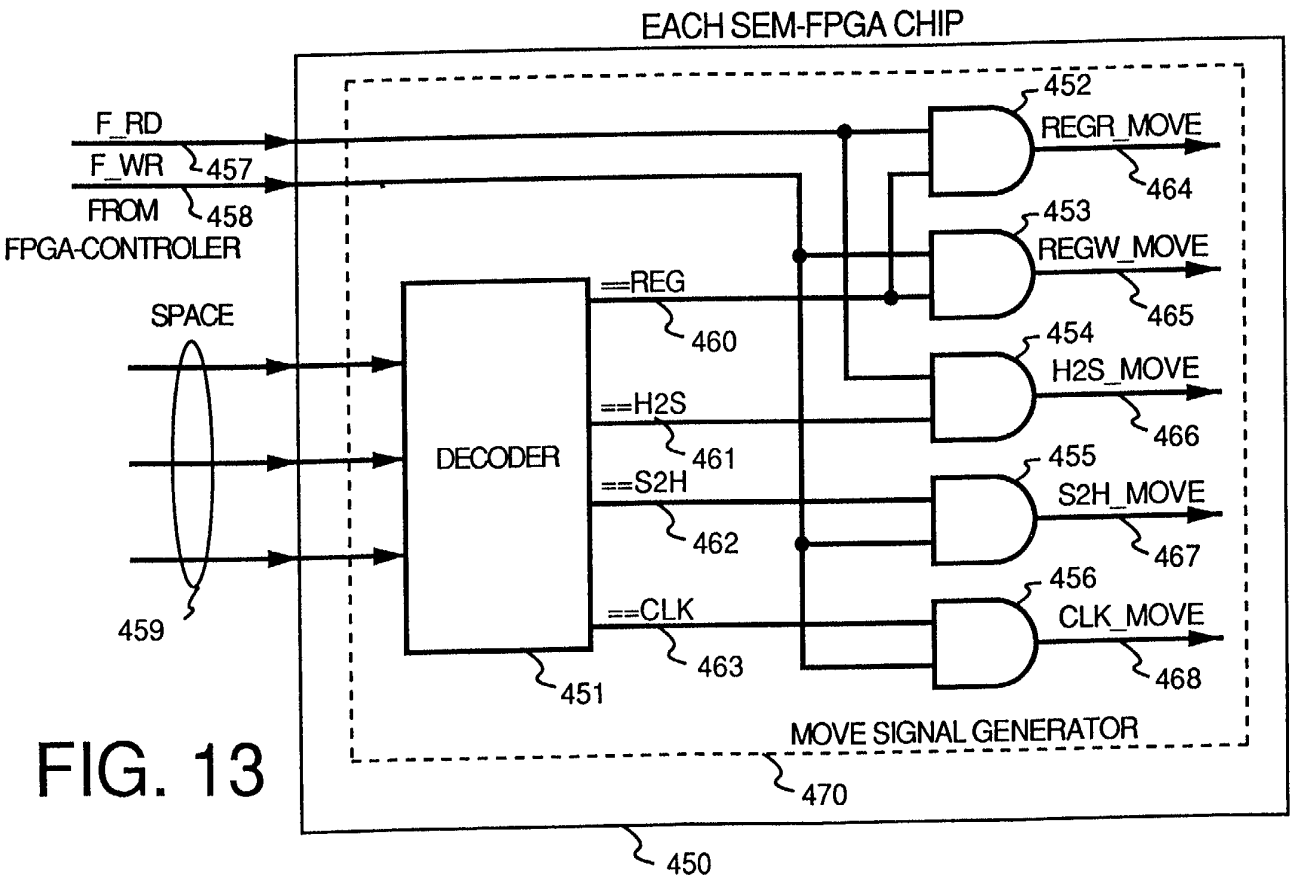


FIG. 13

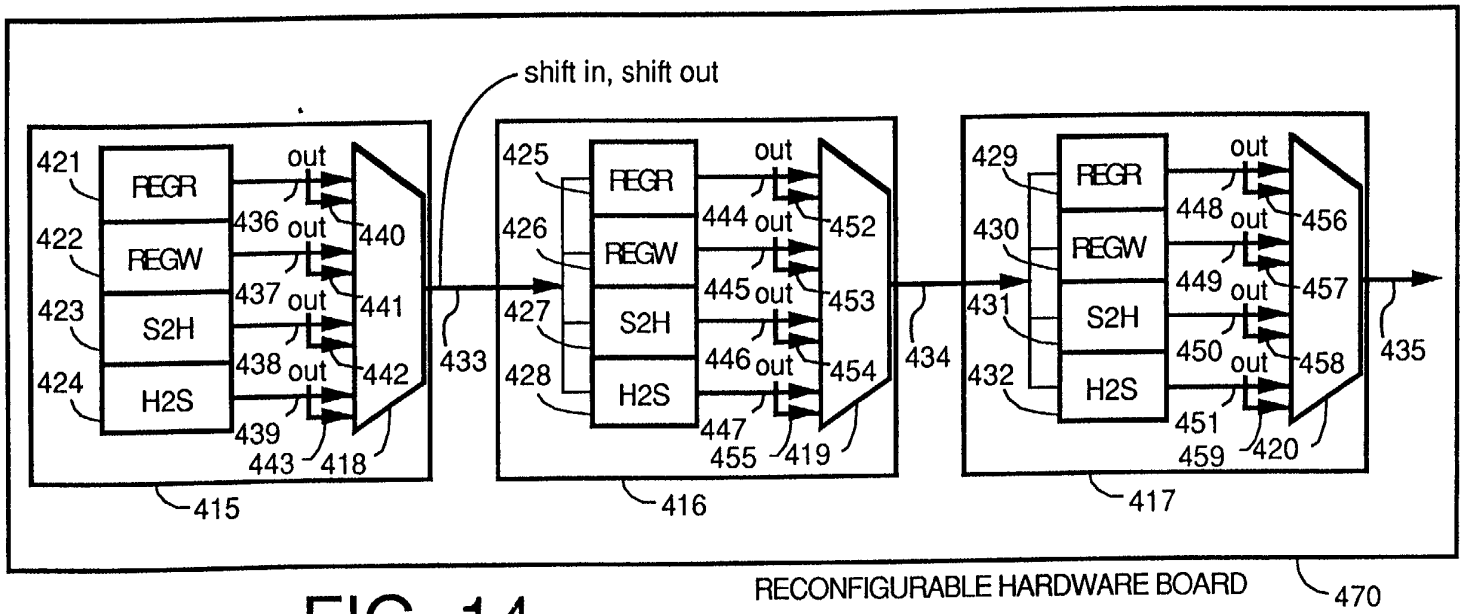


FIG. 14

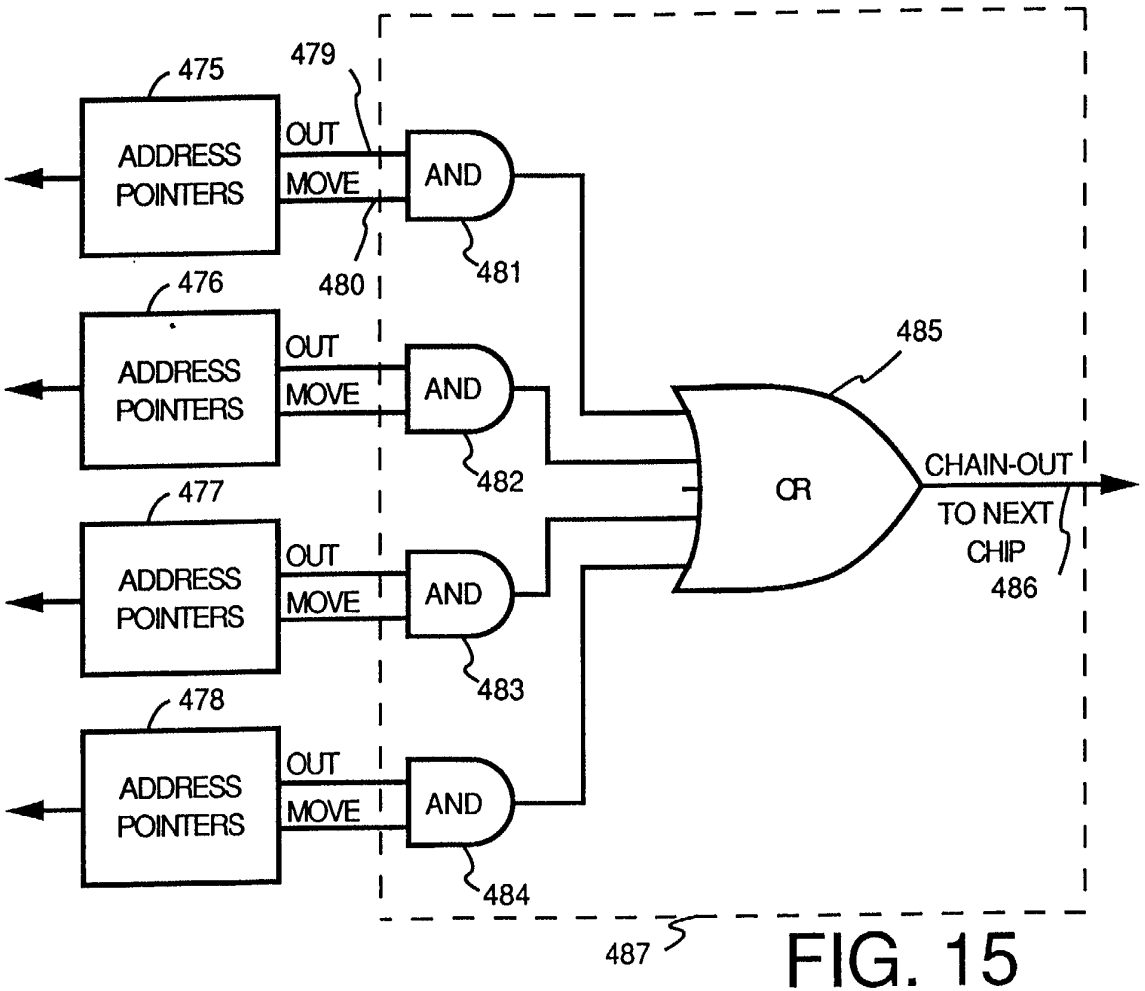


FIG. 15

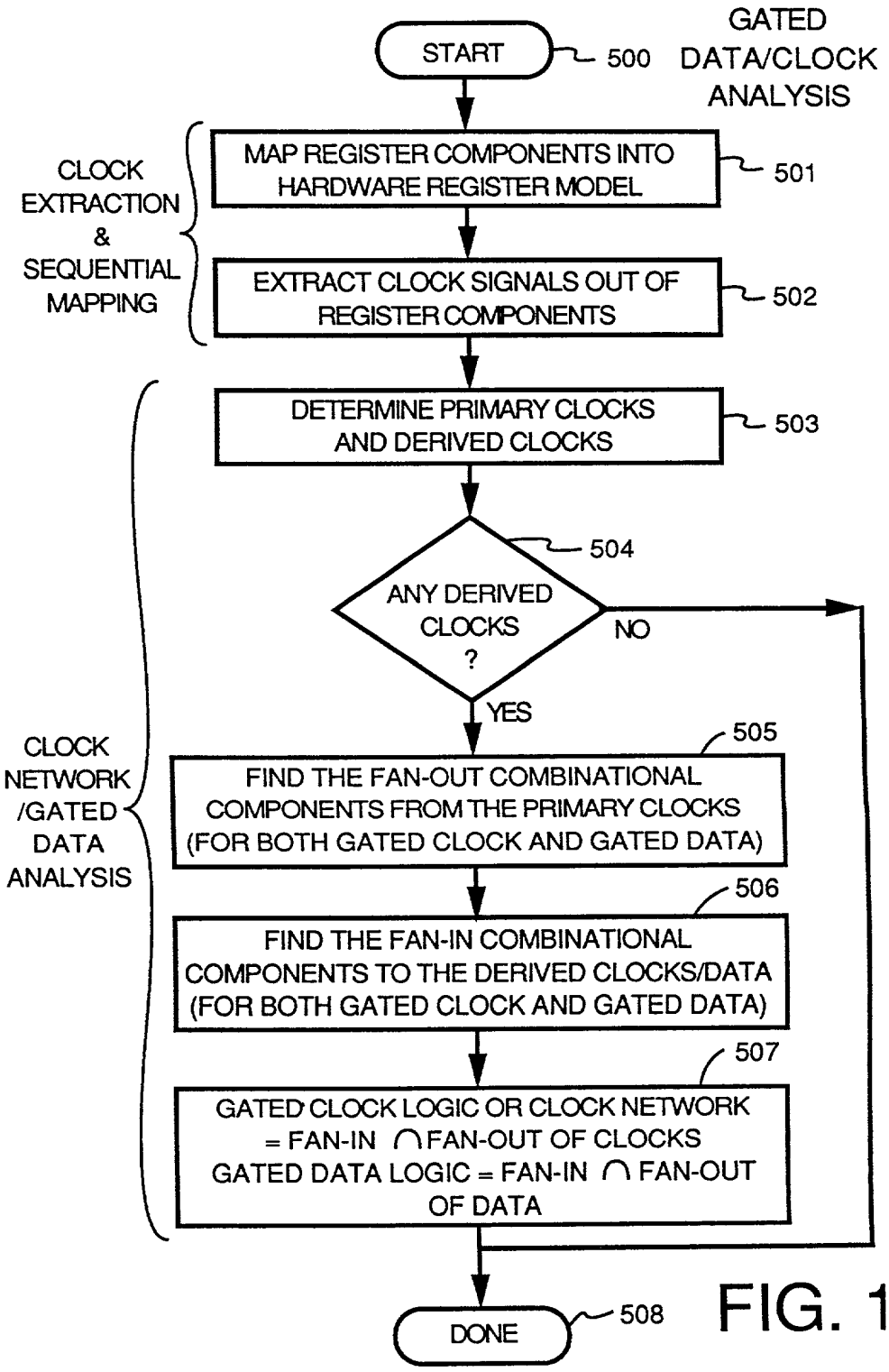

GATED DATA/CLOCK ANALYSIS
START
500
CLOCK EXTRACTION & SEQUENTIAL MAPPING
MAP REGISTER COMPONENTS INTO HARDWARE REGISTER MODEL
501
EXTRACT CLOCK SIGNALS OUT OF REGISTER COMPONENTS
502
DETERMINE PRIMARY CLOCKS AND DERIVED CLOCKS
503
504
ANY DERIVED CLOCKS ?
NO
YES
CLOCK NETWORK /GATED DATA ANALYSIS
505
FIND THE FAN-OUT COMBINATIONAL COMPONENTS FROM THE PRIMARY CLOCKS (FOR BOTH GATED CLOCK AND GATED DATA)
506
FIND THE FAN-IN COMBINATIONAL COMPONENTS TO THE DERIVED CLOCKS/DATA (FOR BOTH GATED CLOCK AND GATED DATA)
507
GATED CLOCK LOGIC OR CLOCK NETWORK = FAN-IN ∩ FAN-OUT OF CLOCKS
GATED DATA LOGIC = FAN-IN ∩ FAN-OUT OF DATA
DONE
508

FIG. 16

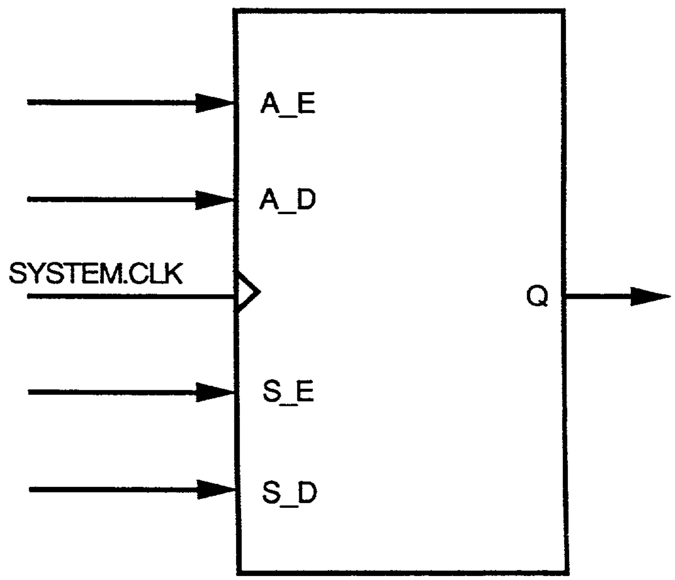


FIG. 17

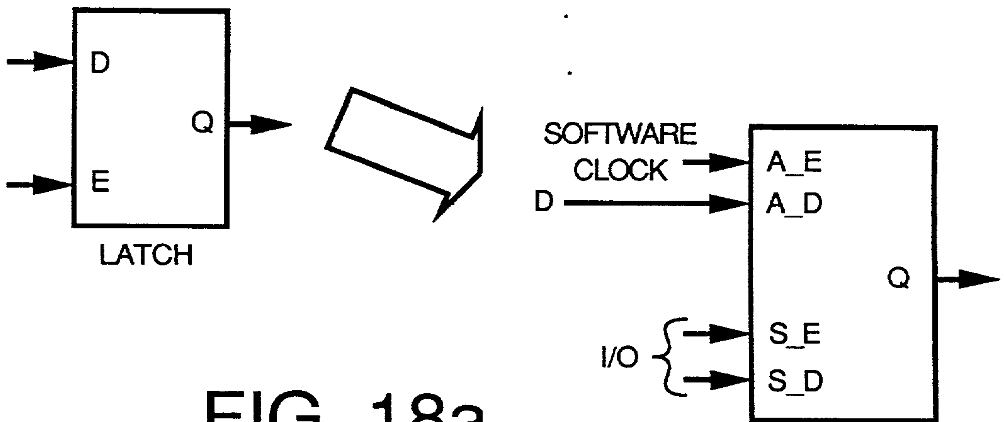


FIG. 18a

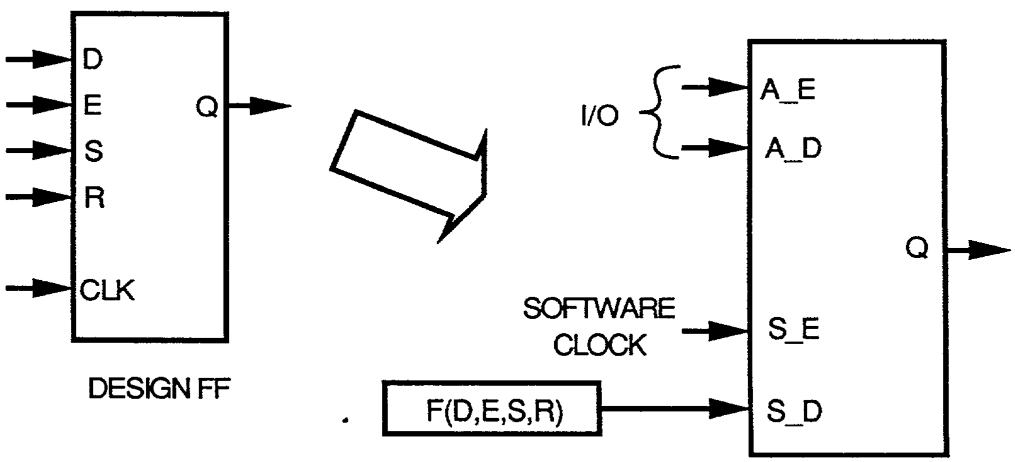


FIG. 18b

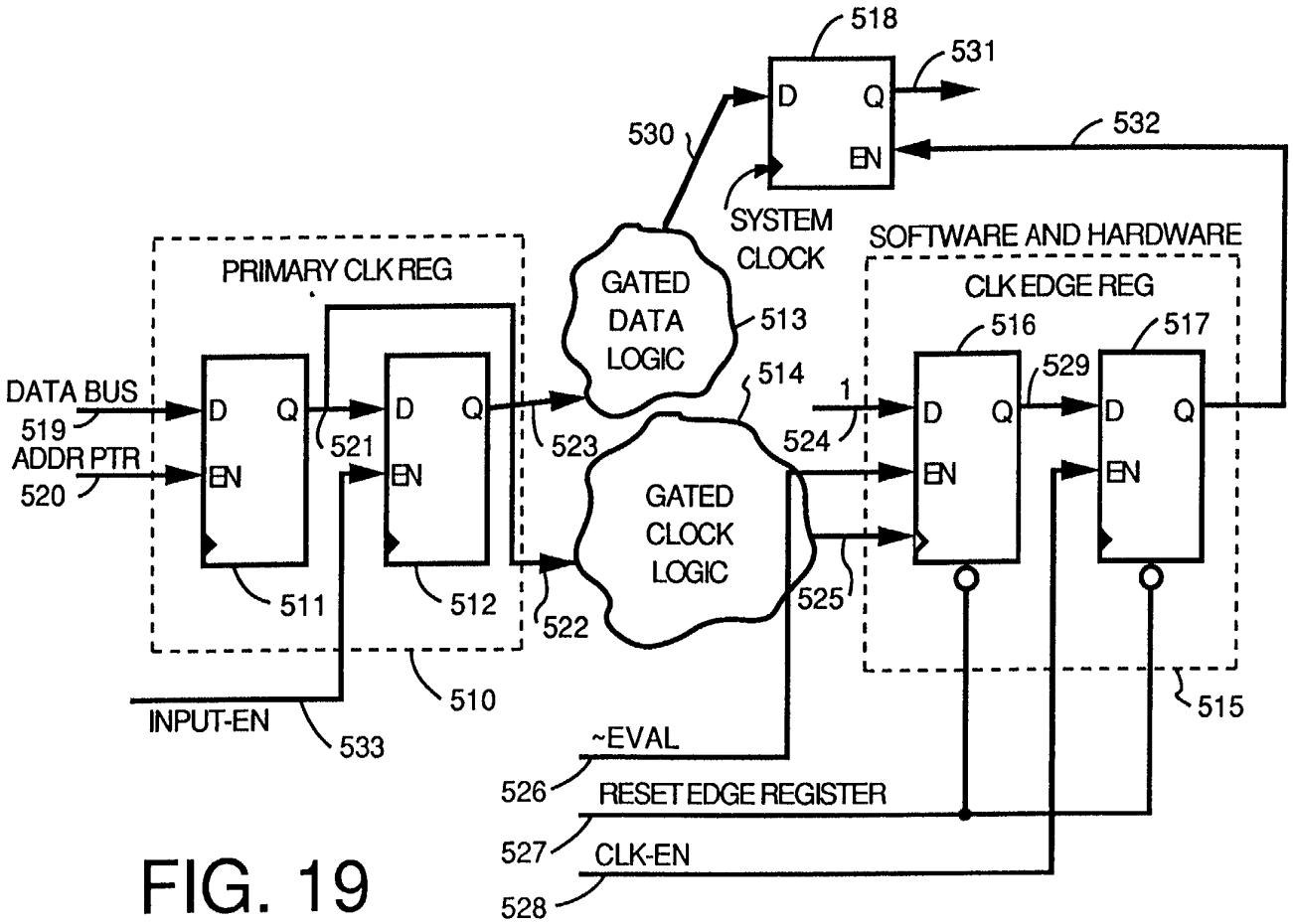


FIG. 19

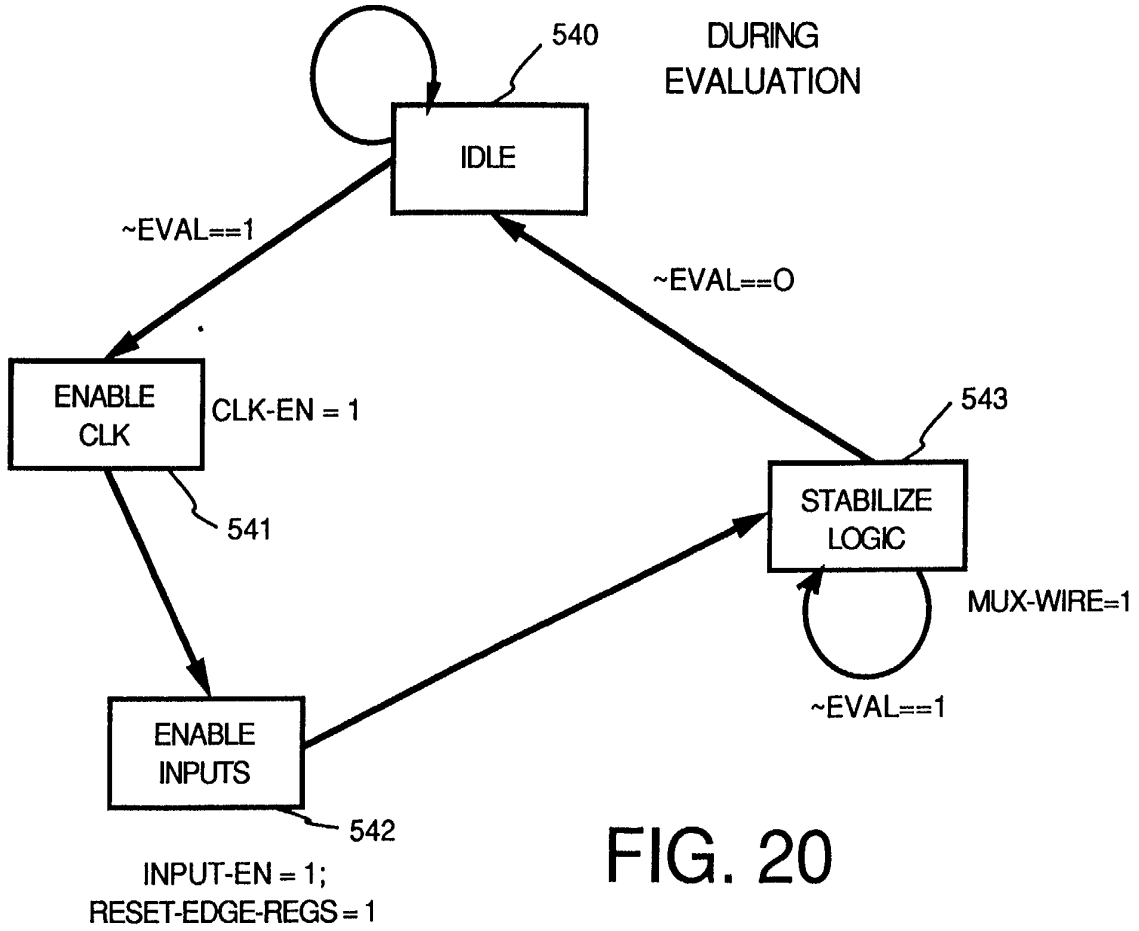


FIG. 20

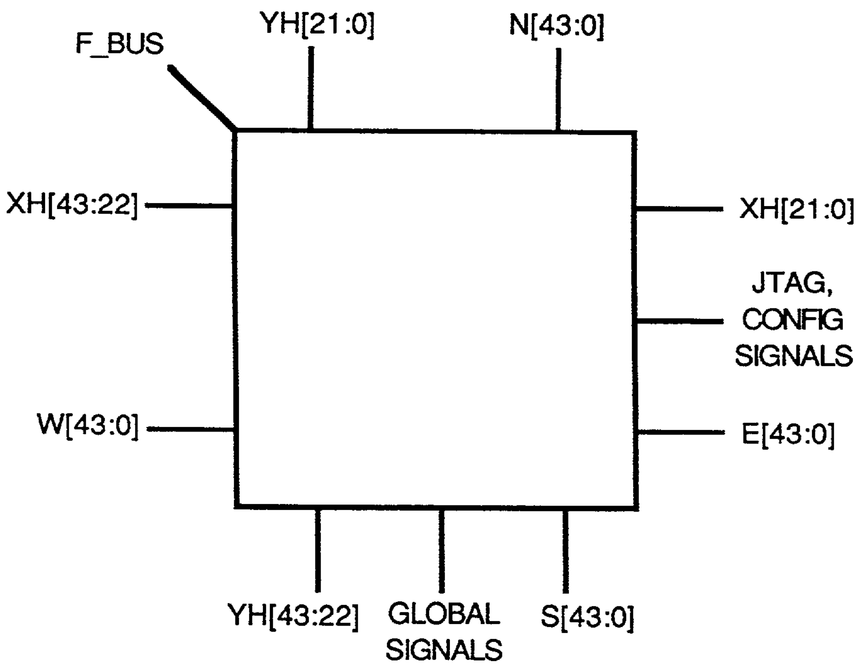

F_BUS
YH[21:0]
N[43:0]
XH[43:22]
XH[21:0]
JTAG,
CONFIG
SIGNALS
W[43:0]
E[43:0]
YH[43:22]
GLOBAL
SIGNALS
S[43:0]

FIG. 21

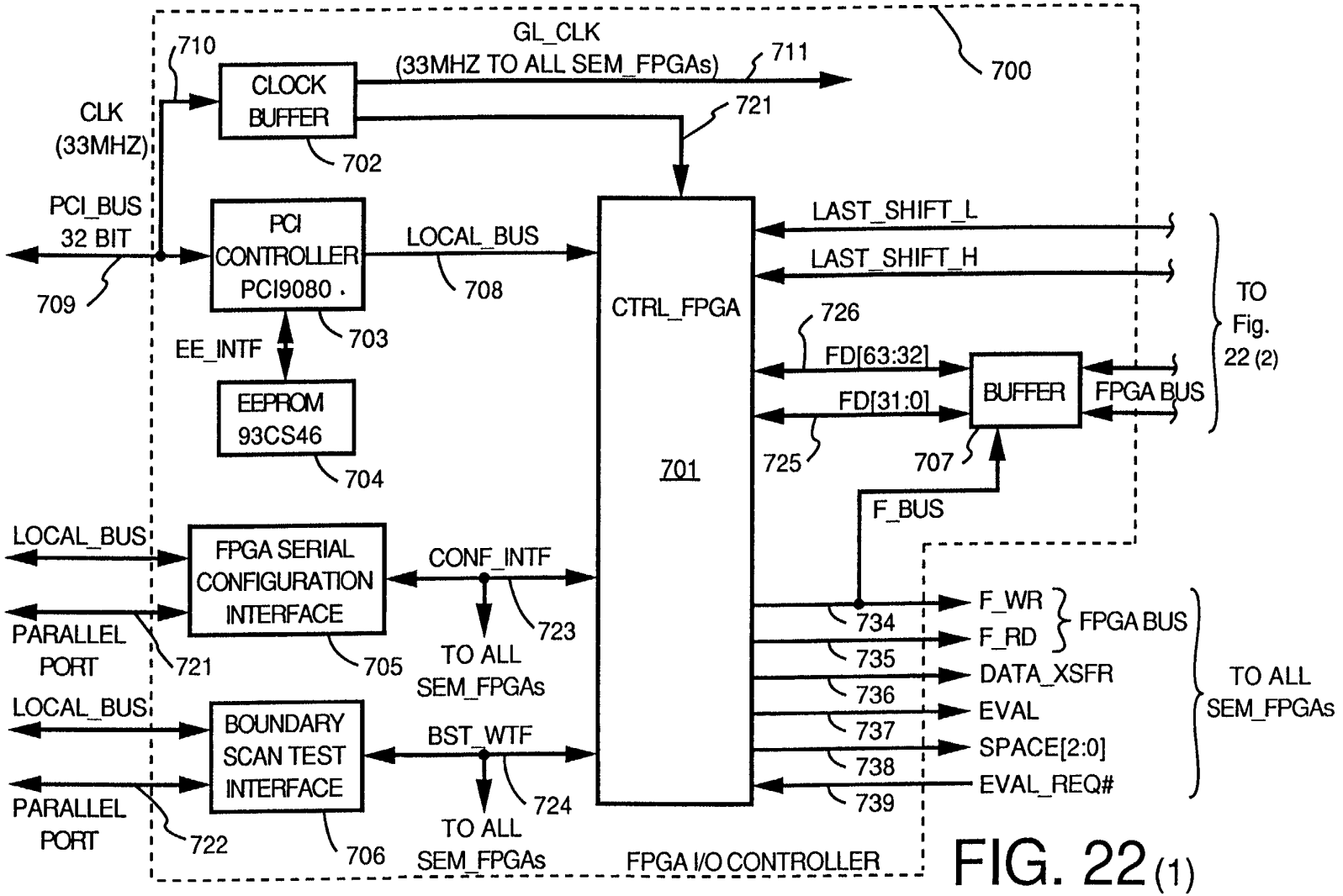


FIG. 22 (1)

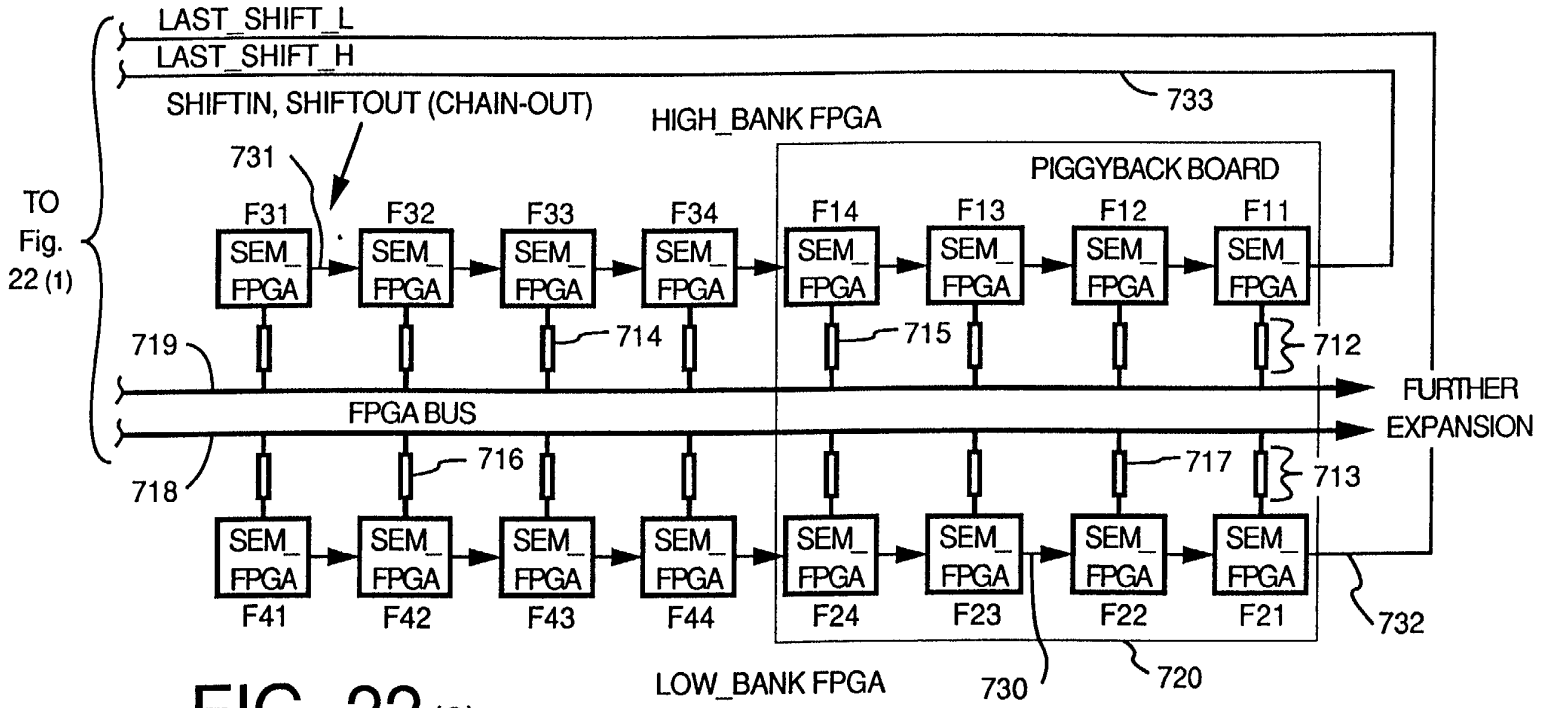

LAST_SHIFT_L
LAST_SHIFT_H
SHIFTIN, SHIFTOUT (CHAIN-OUT)
733
HIGH_BANK FPGA
731
PIGGYBACK BOARD
TO
Fig.
22 (1)
F31
F32
F33
F34
F14
F13
F12
F11
SEM_
FPGA
719
714
715
712
FURTHER
EXPANSION
FPGA BUS
718
716
717
713
F41
F42
F43
F44
F24
F23
F22
F21
732
730
720
LOW_BANK FPGA

FIG. 22 (2)

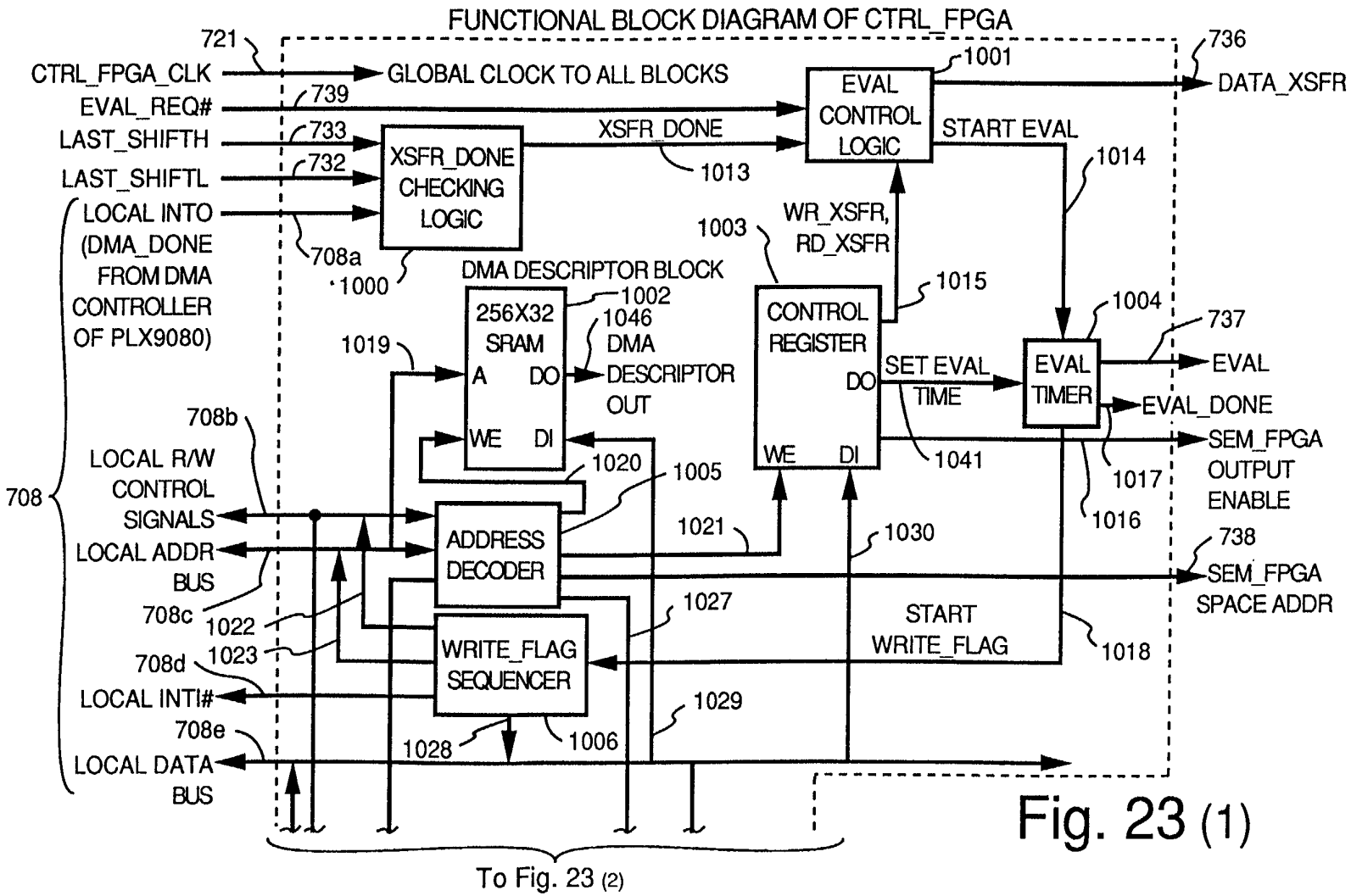

FUNCTIONAL BLOCK DIAGRAM OF CTRL_FPGA
721
CTRL_FPGA_CLK
GLOBAL CLOCK TO ALL BLOCKS
EVAL_REQ#
739
LAST_SHIFTH
733
LAST_SHIFTL
732
LOCAL INTO (DMA_DONE FROM DMA CONTROLLER OF PLX9080)
708a
XSFR_DONE CHECKING LOGIC
1000
XSFR_DONE
1013
EVAL CONTROL LOGIC
1001
START EVAL
1014
736
DATA_XSFR
1003
WR_XSFR, RD_XSFR
1015
DMA DESCRIPTOR BLOCK
256X32 SRAM
1002
1046
DMA DESCRIPTOR OUT
A
DO
WE
DI
1019
CONTROL REGISTER
1004
EVAL TIMER
737
EVAL
SET EVAL TIME
EVAL_DONE
SEM_FPGA OUTPUT ENABLE
1041
1017
1016
708b
LOCAL R/W CONTROL SIGNALS
708
LOCAL ADDR BUS
708c
1022
1020
1005
ADDRESS DECODER
1021
1030
738
SEM_FPGA SPACE ADDR
1027
START WRITE_FLAG
1018
WRITE_FLAG SEQUENCER
1023
708d
LOCAL INTI#
1029
1028
1006
708e
LOCAL DATA BUS
To Fig. 23 (2)

Fig. 23 (1)

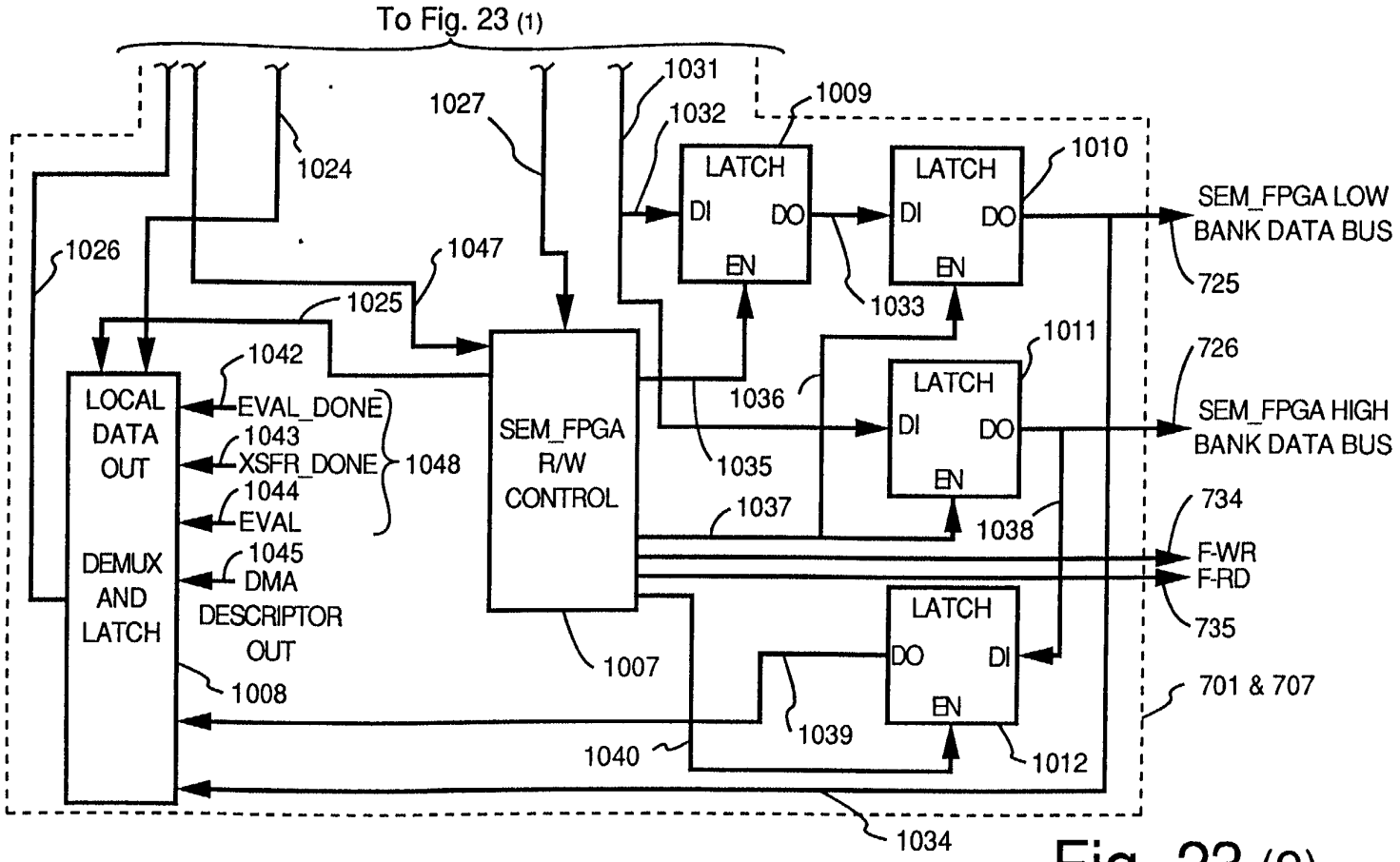


Fig. 23 (2)

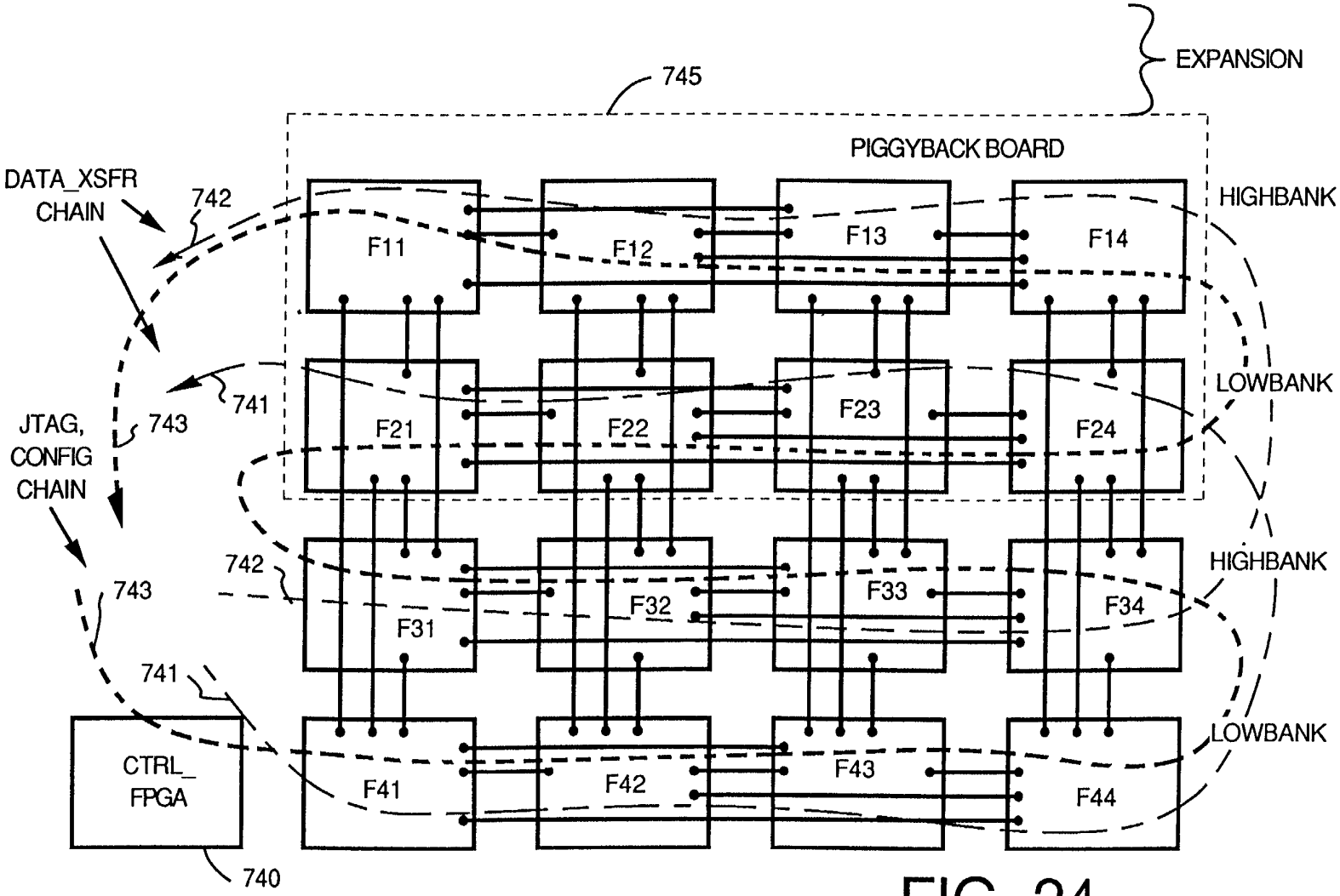


FIG. 24

HARDWARE START-UP

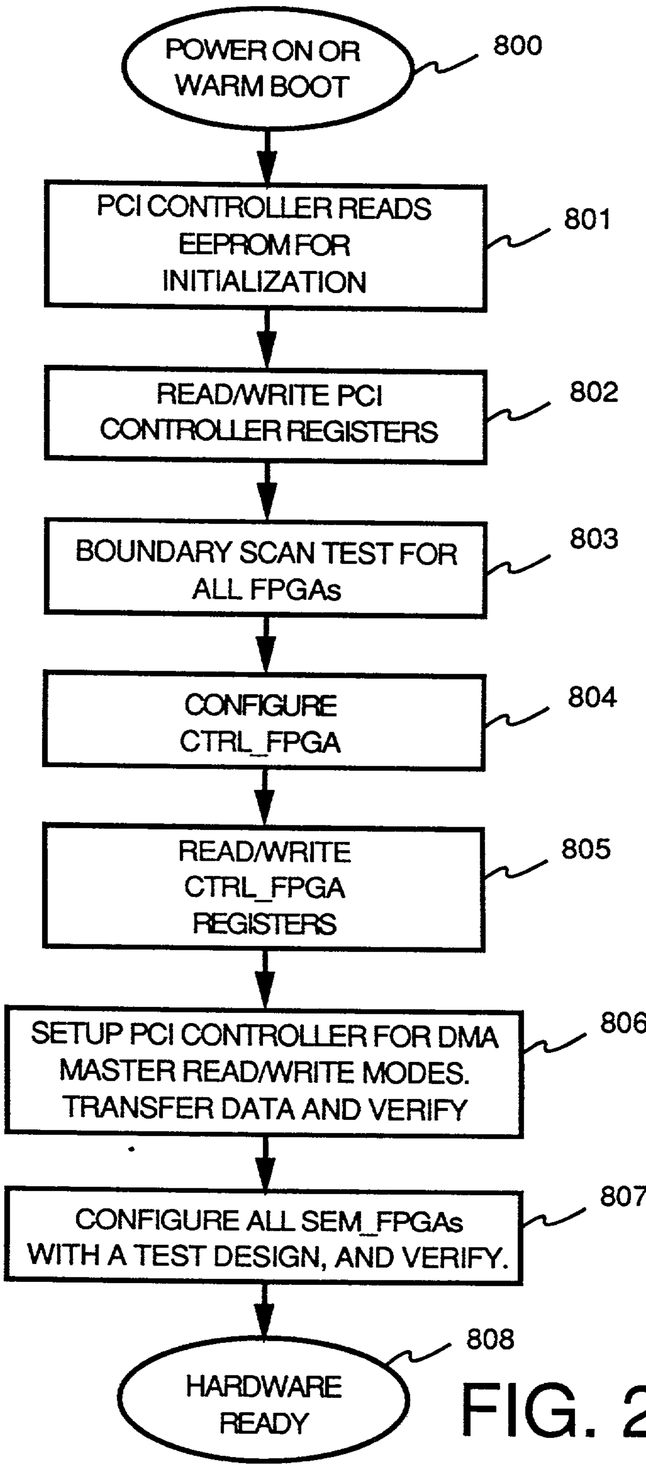


FIG. 25

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(posedge clock or negedge reset)
   if(~reset)
      q = 0;
   else
      q = d;
endmodule

module example;
   wire d1, d2, d3;
   wire q1, q2, q3;
   reg sigin;
   wire sigout;
   reg clk, reset;

   register reg1 (clk, reset, d1, q1);
   register reg2 (clk, reset, d2, q2);
   register reg3 (clk, reset, d3, q3);

   assign d1 = sigin ^ q3;
   assign d2 = q1 ^ q3;
   assign d3 = q2 ^ q3;
   assign sigout = q3;

   // a clock generator
   always
   begin
      clk = 0;
      #5;
      clk = 1;
      #5;
   end

   // a signal generator
   always
   begin
      #10;
      sigin = $random;
   end

   // initialization
   initial
   begin
      reset = 0;
      sigin = 0;
      #1;
      reset =1;
      #5;
      $monitor($time, " %b, %b", sigin, sigout);
      #1000 $finish;
   end
   end module
```

Fig. 26

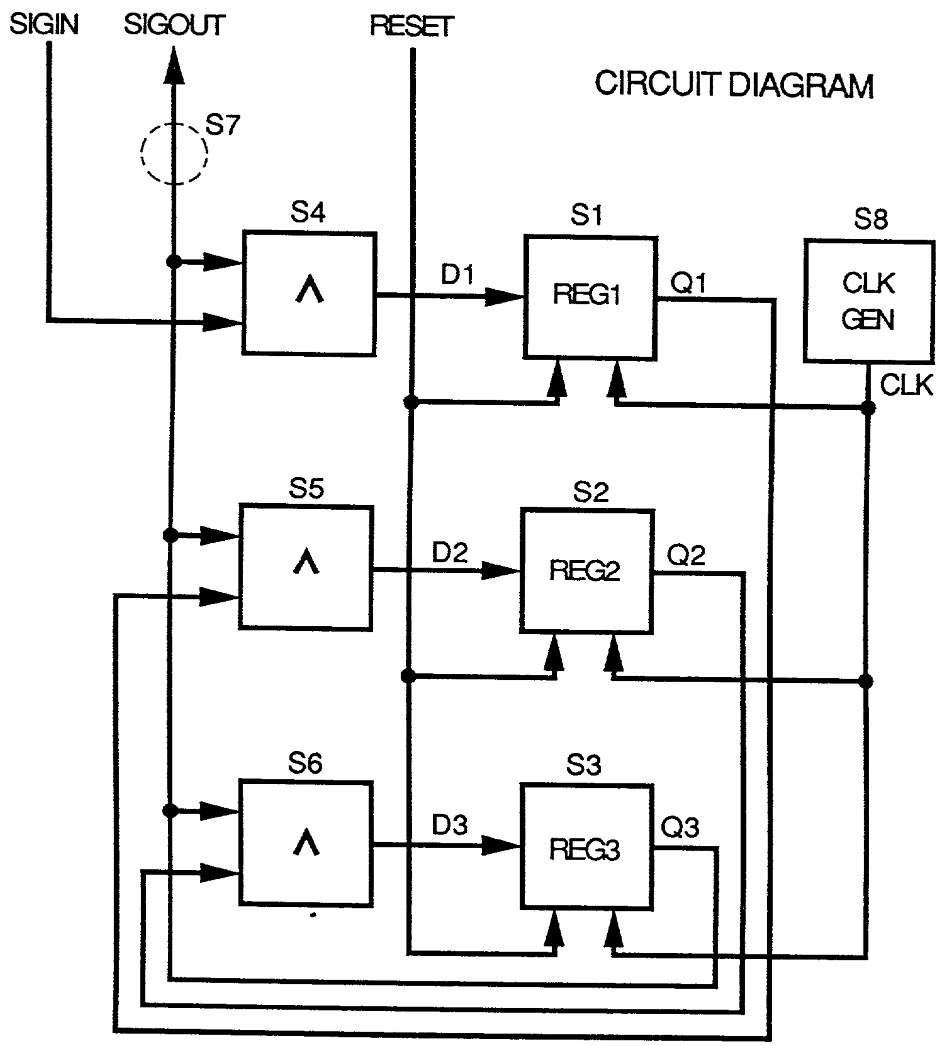

SIGIN
SIGOUT
RESET
CIRCUIT DIAGRAM
S7
S4
S1
S8
D1
REG1
Q1
CLK
GEN
CLK
S5
S2
D2
REG2
Q2
S6
S3
D3
REG3
Q3

FIG. 27

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(post edge clock or negedge reset)
   if(~reset)
     q = 0
   else
     q = d;

endmodule
```

Register definition

900

```
module example;
   wire d1, d2, d3;
   ware q1, q2, q3;

   reg sigin;
   wire sigout;
   reg clk, reset;
```

Wire interconnection info

907

Test-bench input -- 908

Test-bench output -- 909

```
S1  register reg 1 (clk, reset, d1, q1);
S2  register reg 2 (clk, reset, d2, q2);
S3  register reg 3 (clk, reset, d3, q3);
```

Register component

901

```
S4  assign d1 = sigin ^ q3;
S5  assign d2 = q1 ^ 3;
S6  assign d3 = q2 ^ q3;
S7  assign signout = q3;
```

Combinational component

902

```
S8  // a clock generator
    always
    begin
      clk = 0;
      #5;
      clk = 1;
      #5;
    end
```

Clock component

903

```
S9  // a signal generator
    always
    begin
      #10;
      sigin = $random;
    end
```

Test-bench component (Driver)

904

Fig. 28

```
    // initialization
    initial
S10 begin
      reset = 0;
      sigin = 0;
      #1;
S11   reset = 1;
      #5;
S12   $monitor($time, "%b, %b", sigin, sigout);
      #1000 $finish;
    end
    end module
```

Test-bench component (initialization)

905

Test-bench component (monitor)

906

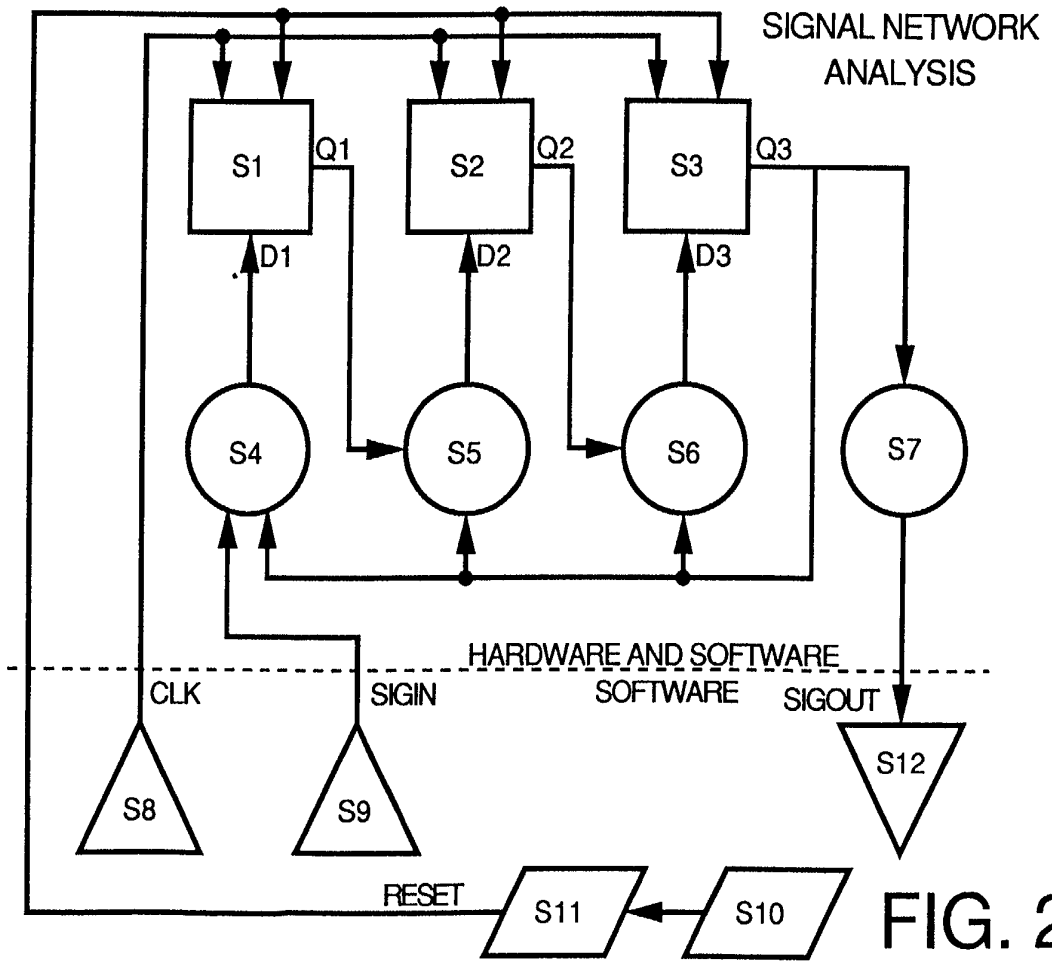


FIG. 29

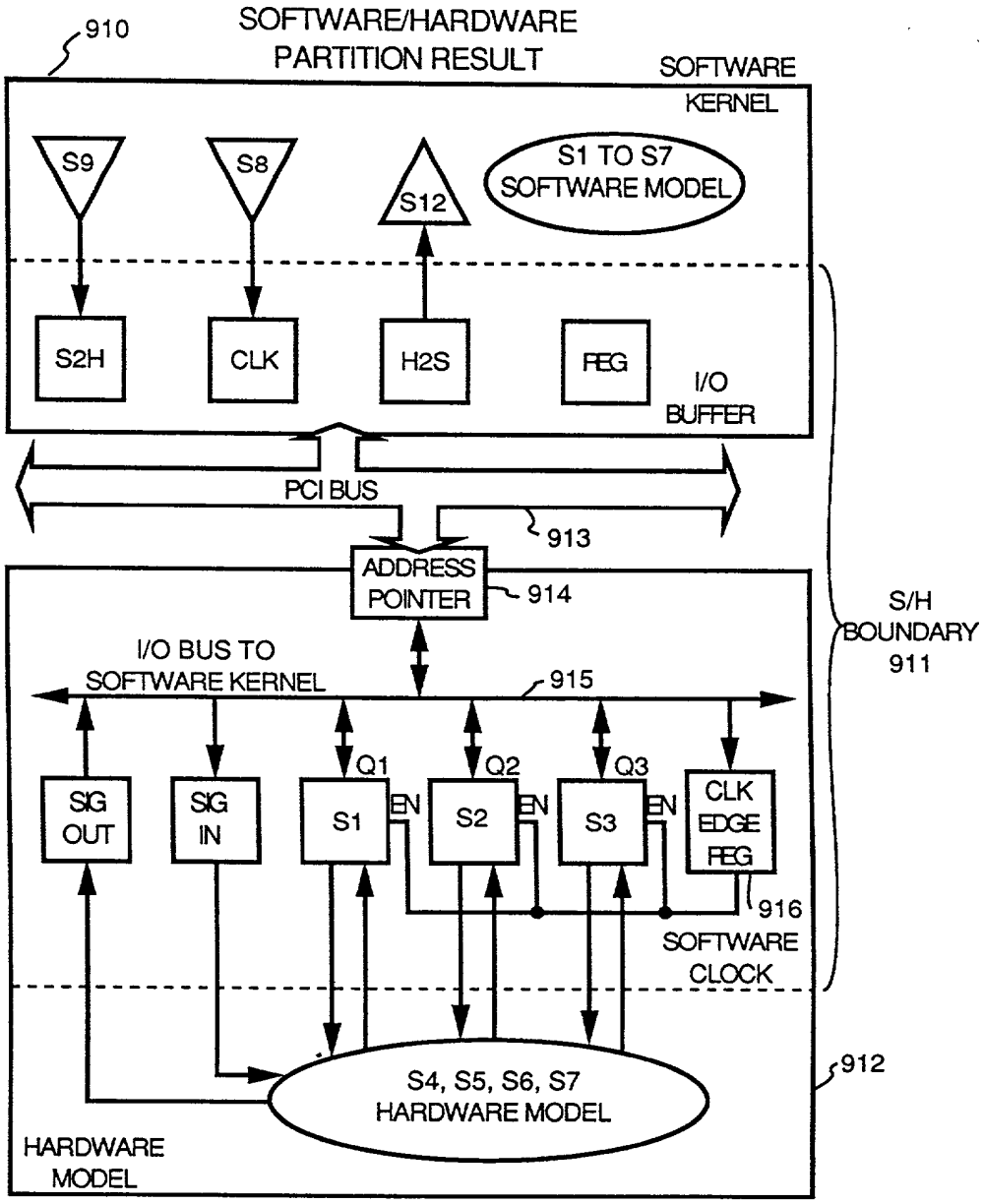

910
SOFTWARE/HARDWARE
PARTITION RESULT
SOFTWARE
KERNEL
S9
S8
S12
S1 TO S7
SOFTWARE MODEL
S2H
CLK
H2S
REG
I/O
BUFFER
PCI BUS
913
ADDRESS
POINTER
914
S/H
BOUNDARY
911
I/O BUS TO
SOFTWARE KERNEL
915
Q1
Q2
Q3
SIG
OUT
SIG
IN
S1
EN
S2
EN
S3
EN
CLK
EDGE
REG
916
SOFTWARE
CLOCK
912
S4, S5, S6, S7
HARDWARE MODEL
HARDWARE
MODEL

FIG. 30

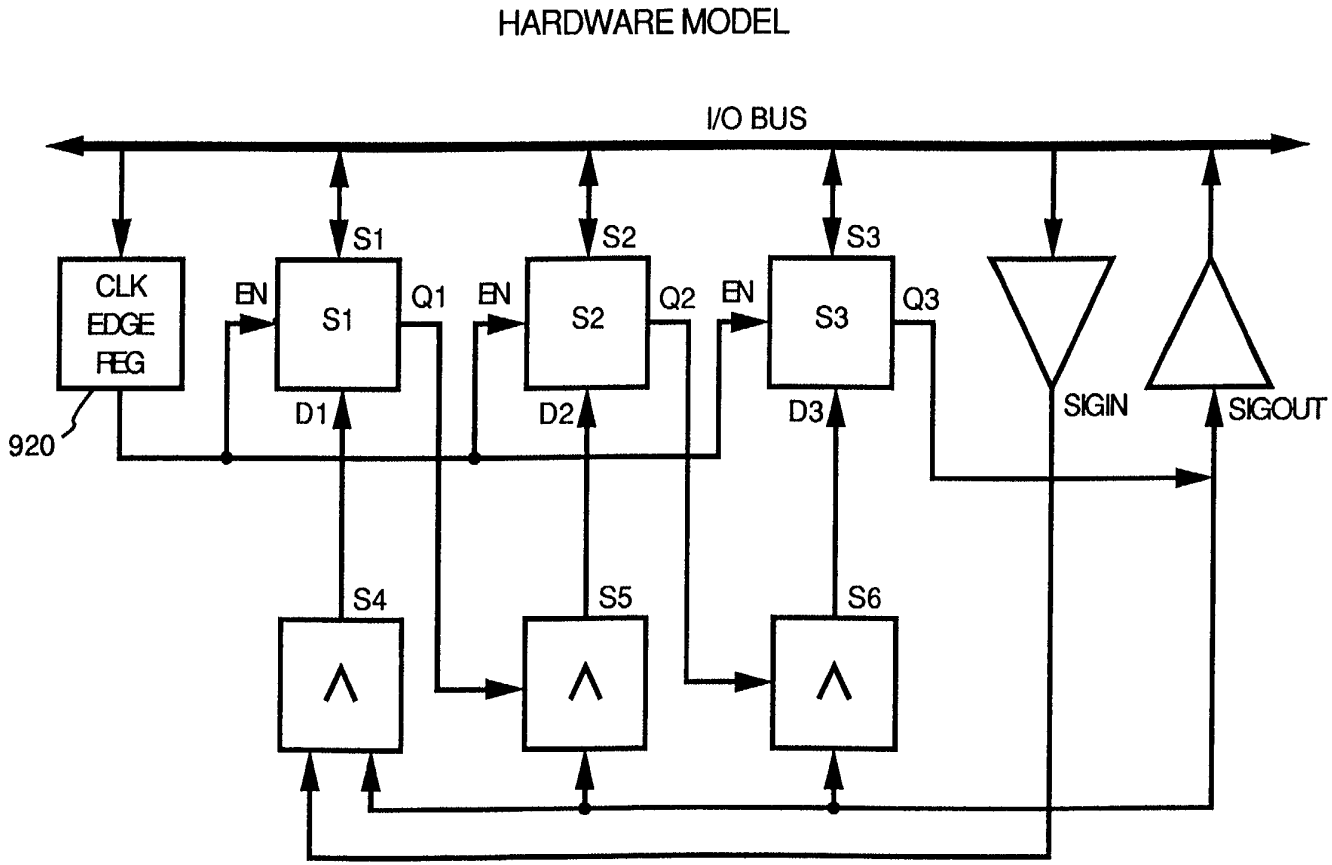
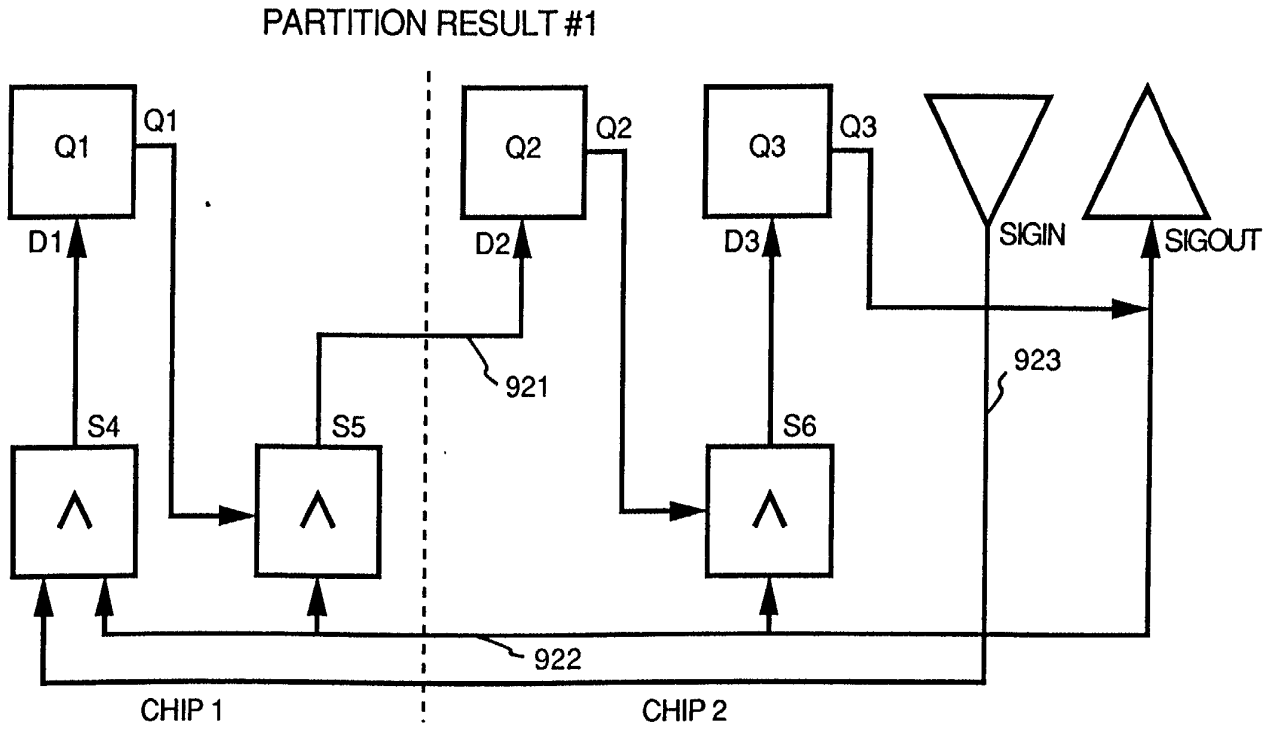


FIG. 31



(IGNORE I/O AND CLOCK EDGE REGISTER)

FIG. 32

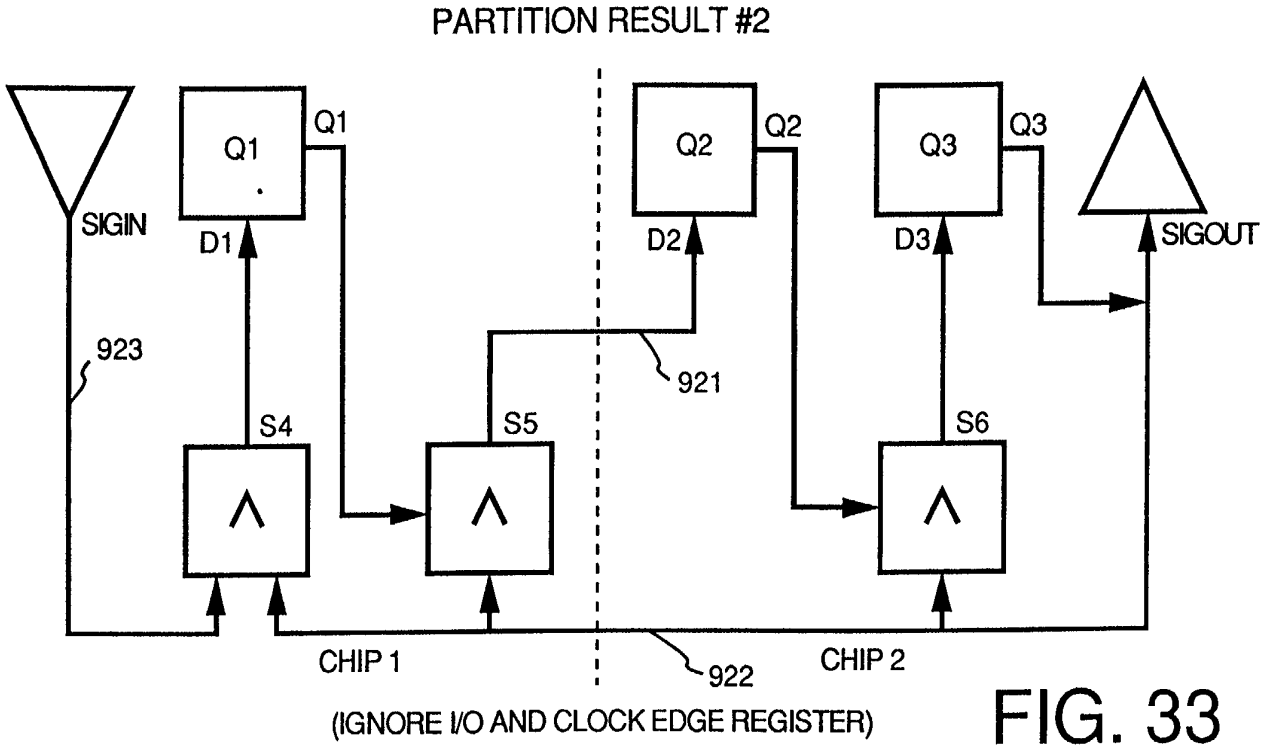


FIG. 33

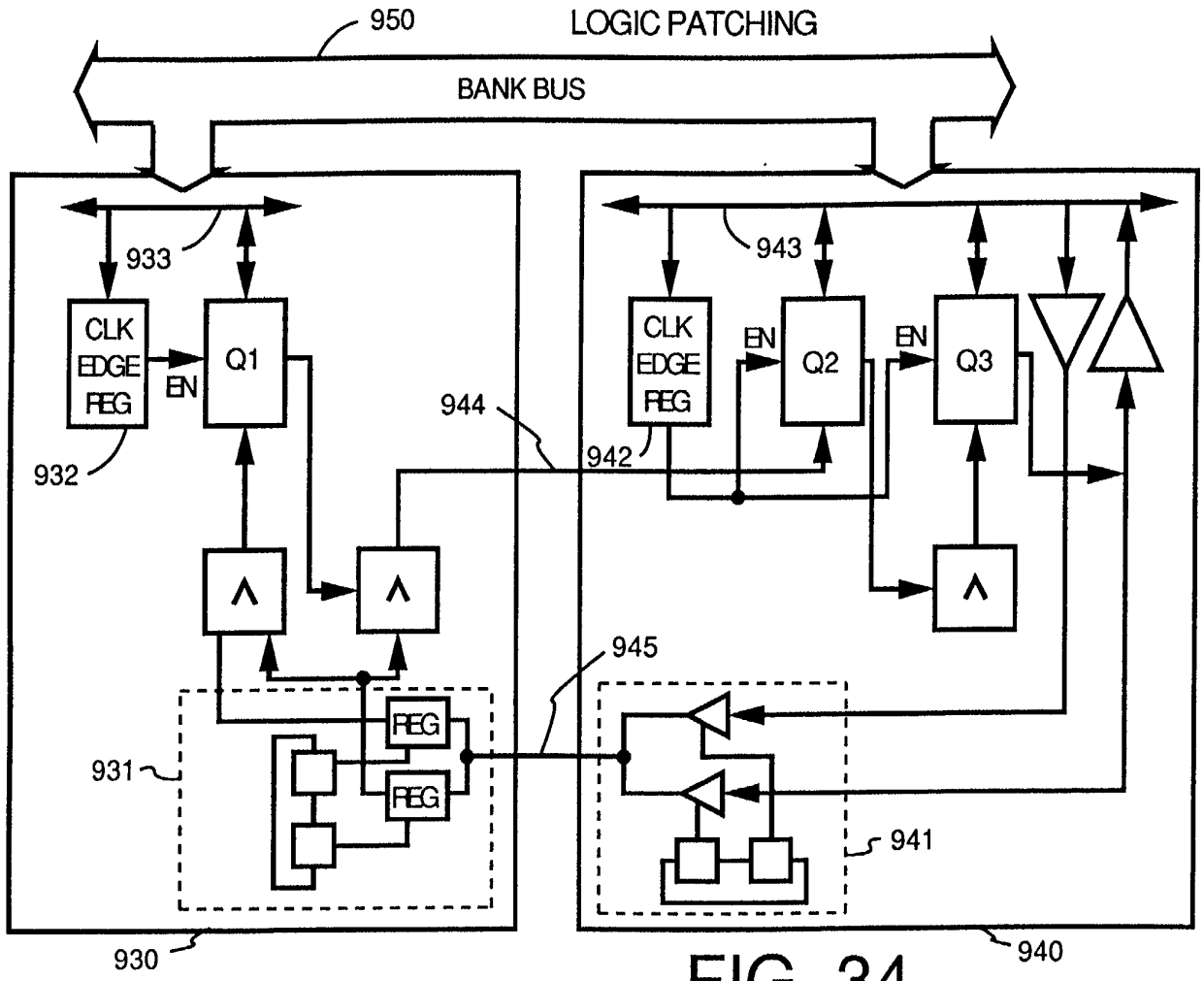


FIG. 34

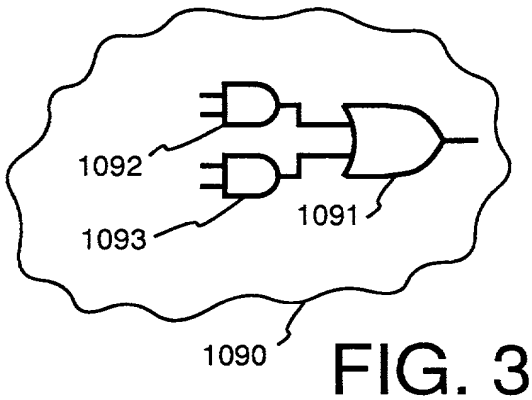


FIG. 35a

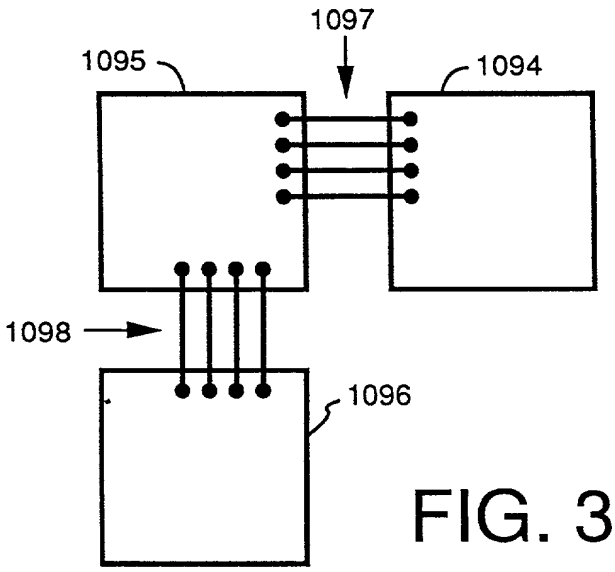


FIG. 35b

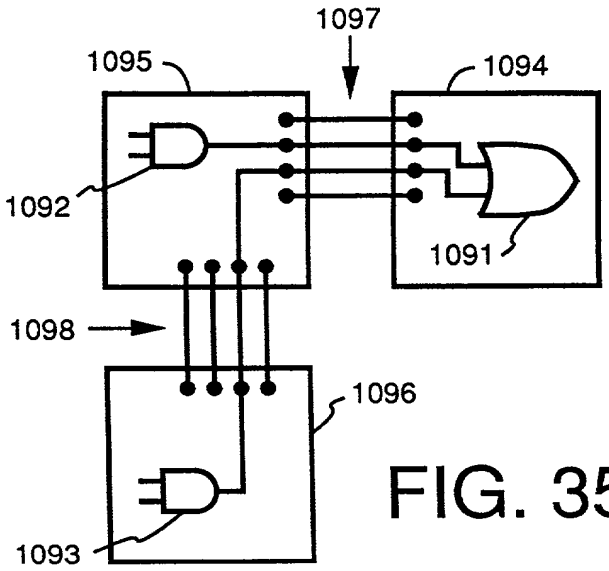


FIG. 35c

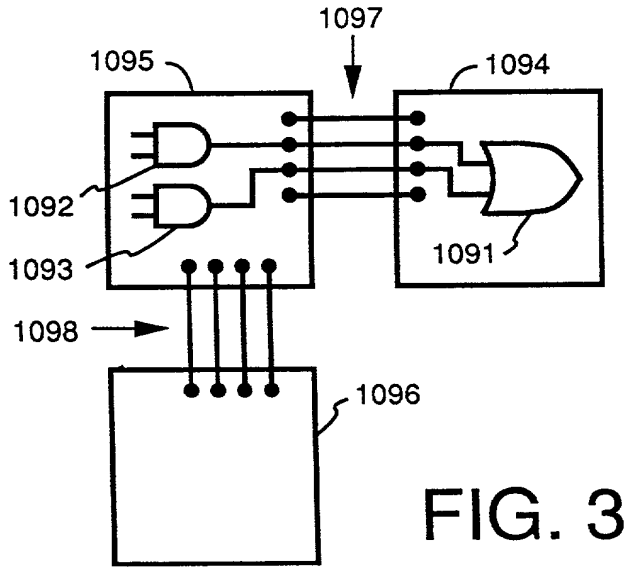
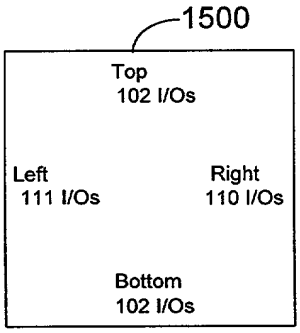


FIG. 35d

I/O PIN OVERVIEW OF FPGA LOGIC DEVICE

FPGA : 10K130V, 10K250V with 599-pin PGA package



425 Interconnect I/O pins

45 Dedicated I/O pins:

GCLK, FD_BUS[31..0], F_RD, F_WR, DATAXSFR, SHIFTIN, SHIFTOUT, SPACE[2..0], EVAL, EV_REQ_N, DEV_OE, DEV_CLRN

FIG. 36

FPGA INTERCONNECT BUSES

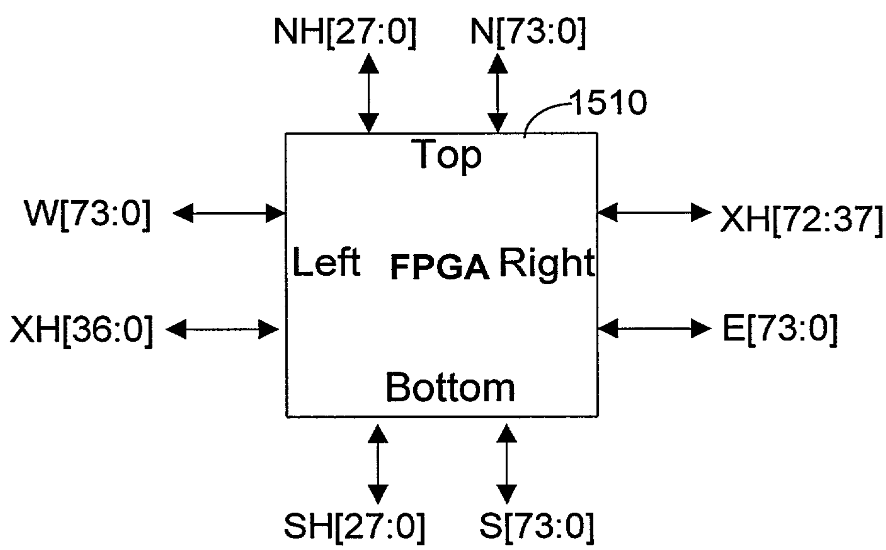


FIG. 37

BOARD CONNECTION - SIDE VIEW

DUAL-BOARD CONFIGURATION

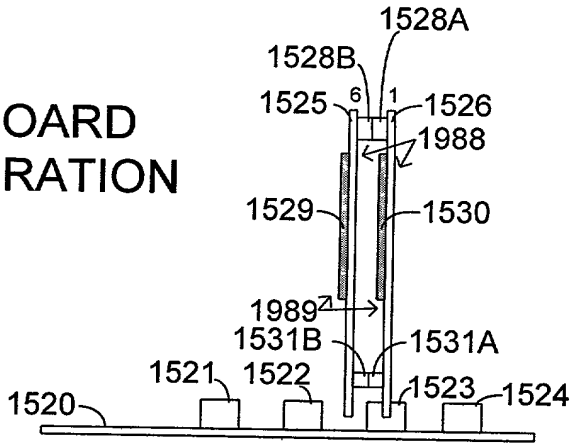


FIG. 38(A)

SIX BOARD CONFIGURATION

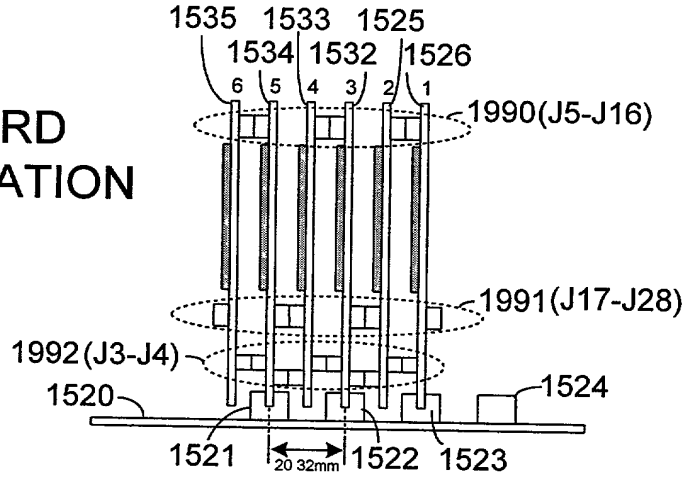


FIG. 38(B)

SIX-BOARD CONFIGURATION DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

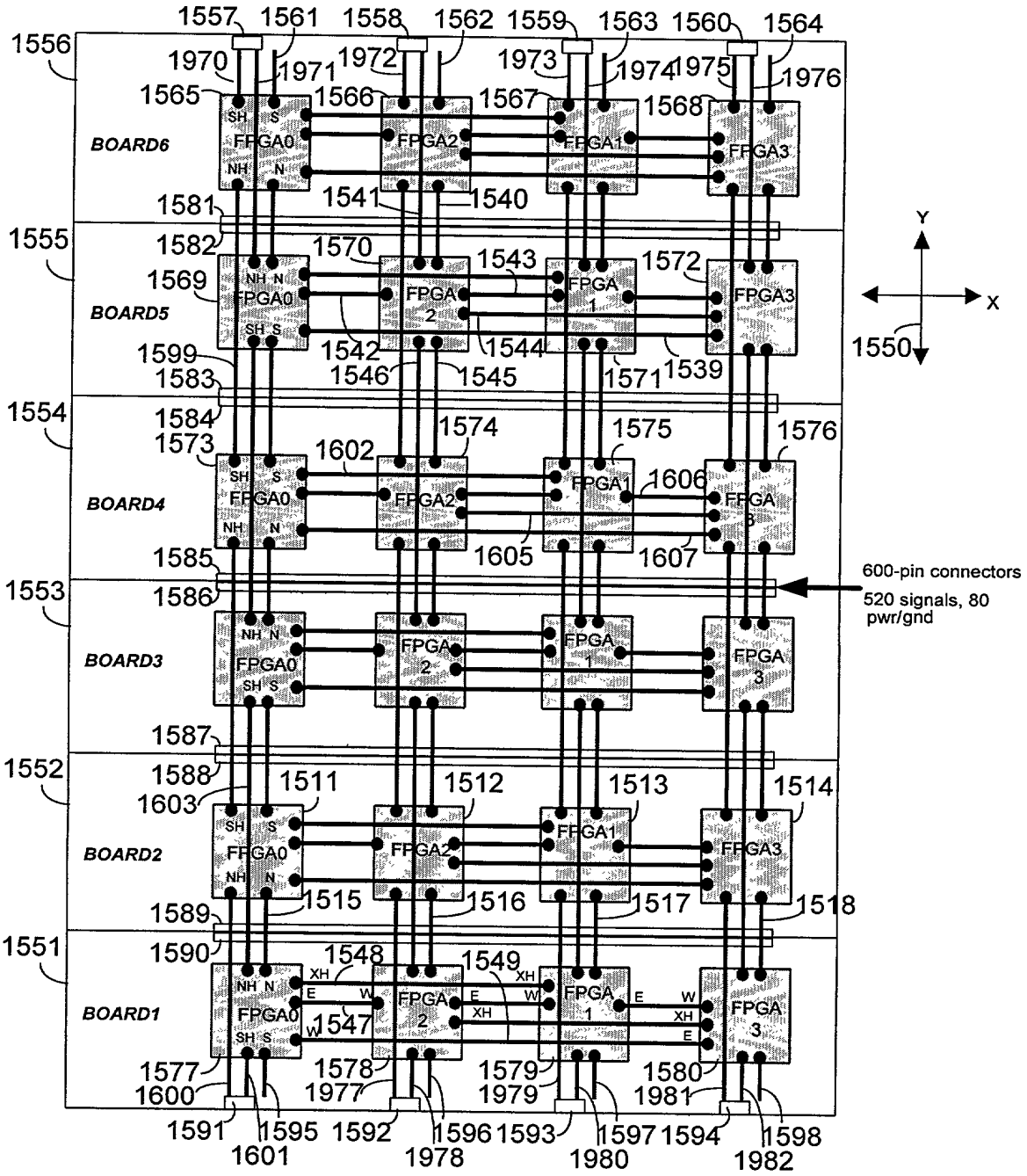


FIG. 39

FPGA ARRAY CONNECTION BETWEEN BOARDS

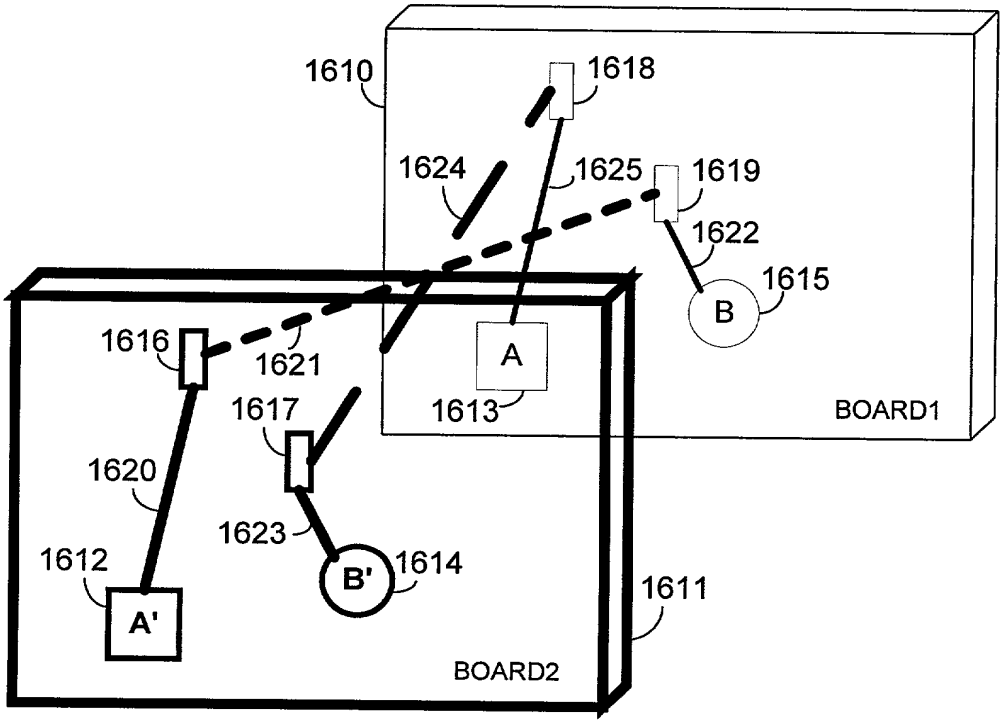


FIG. 40(A)

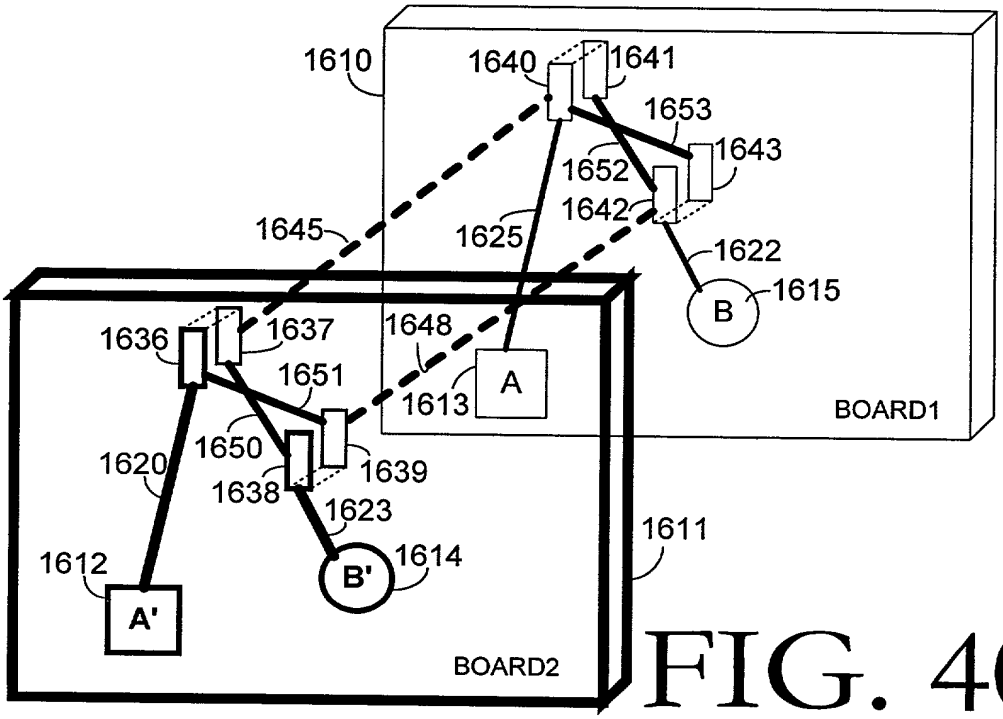


FIG. 40(B)

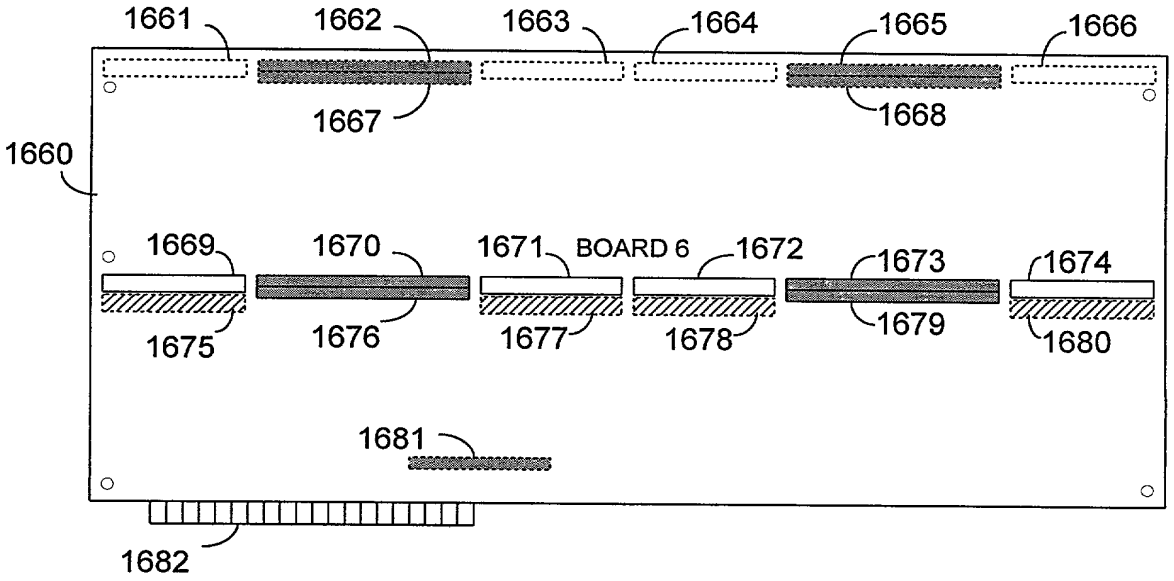


FIG. 41(A)

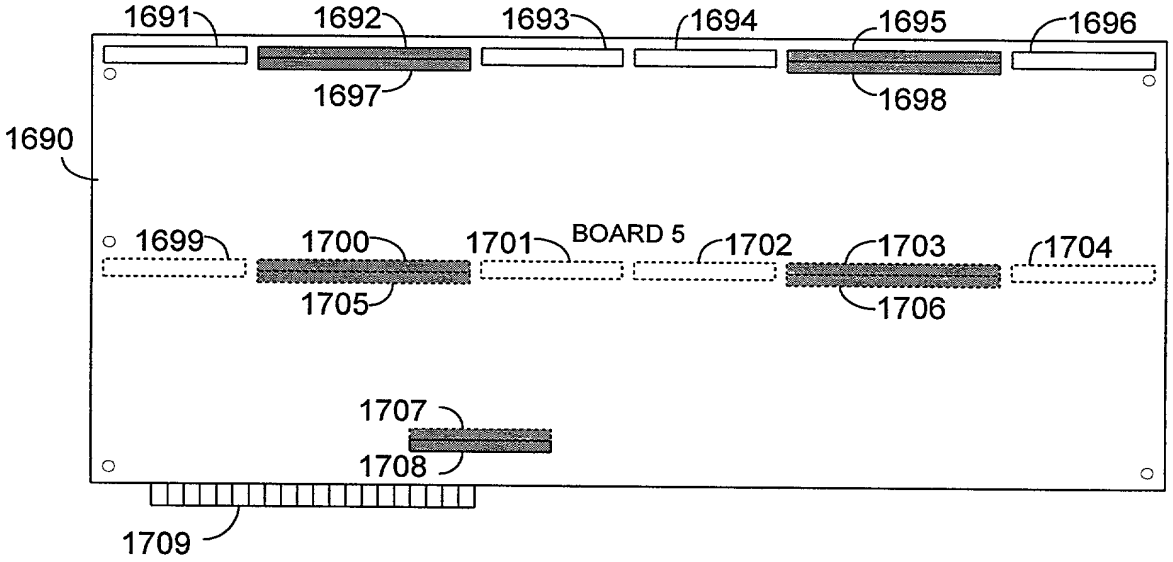


FIG. 41(B)

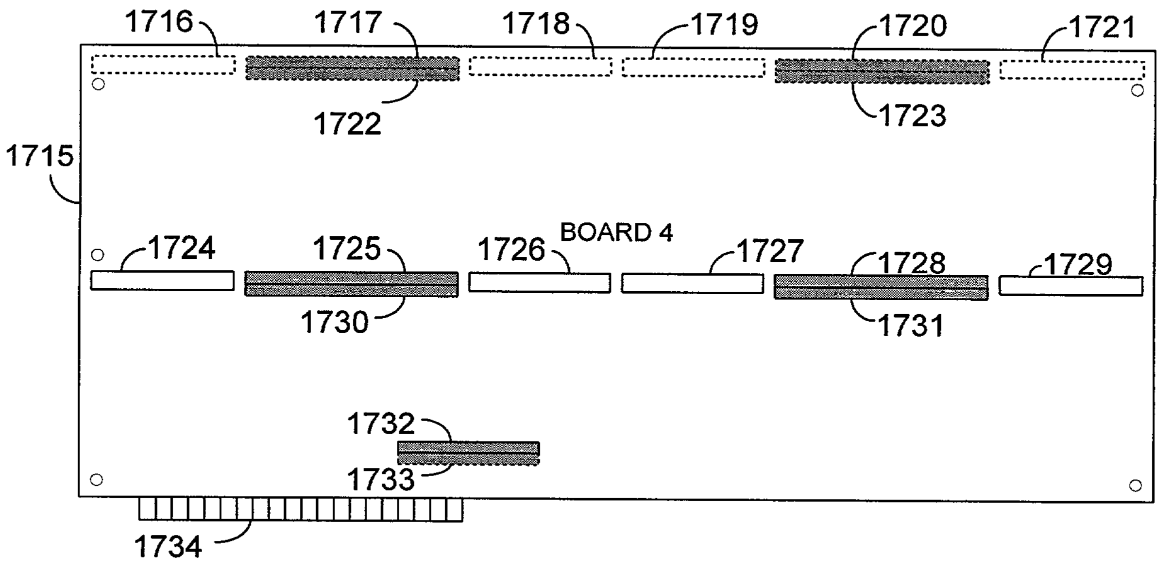


FIG. 41(C)

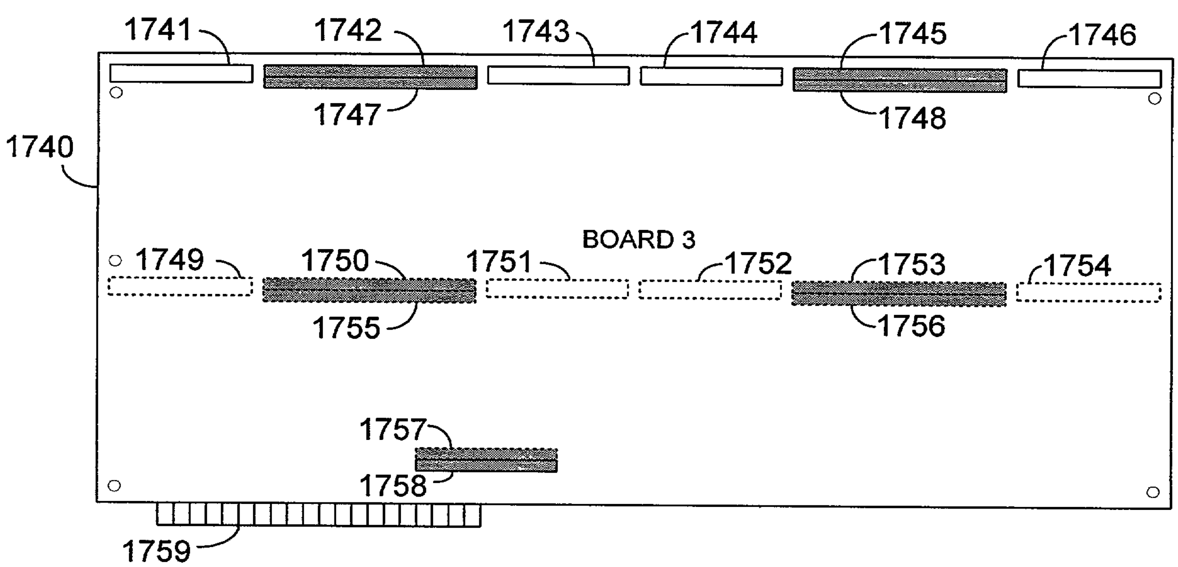


FIG. 41(D)

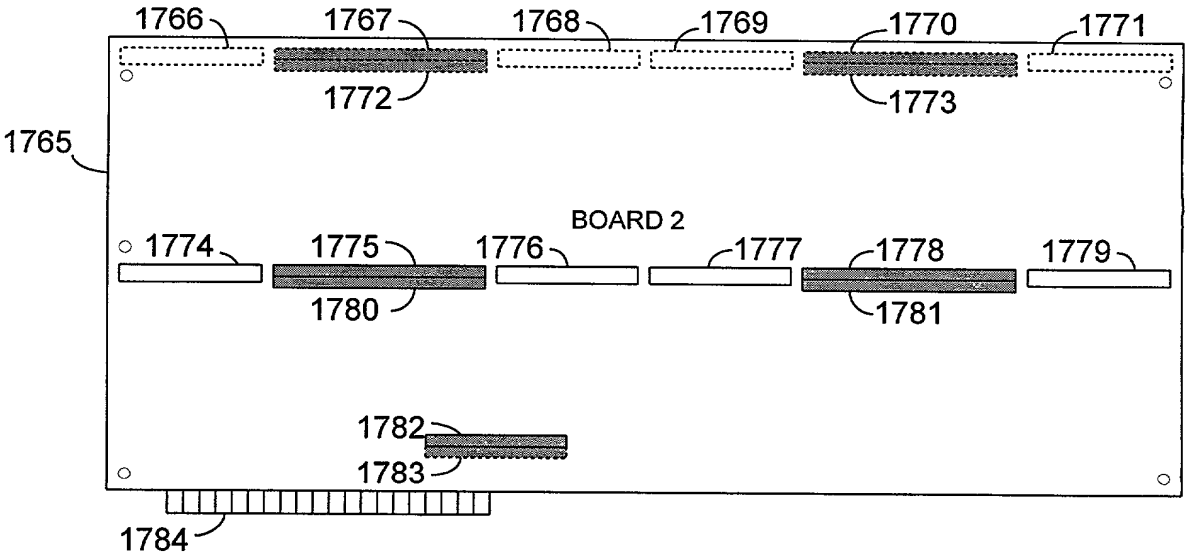


FIG. 41(E)

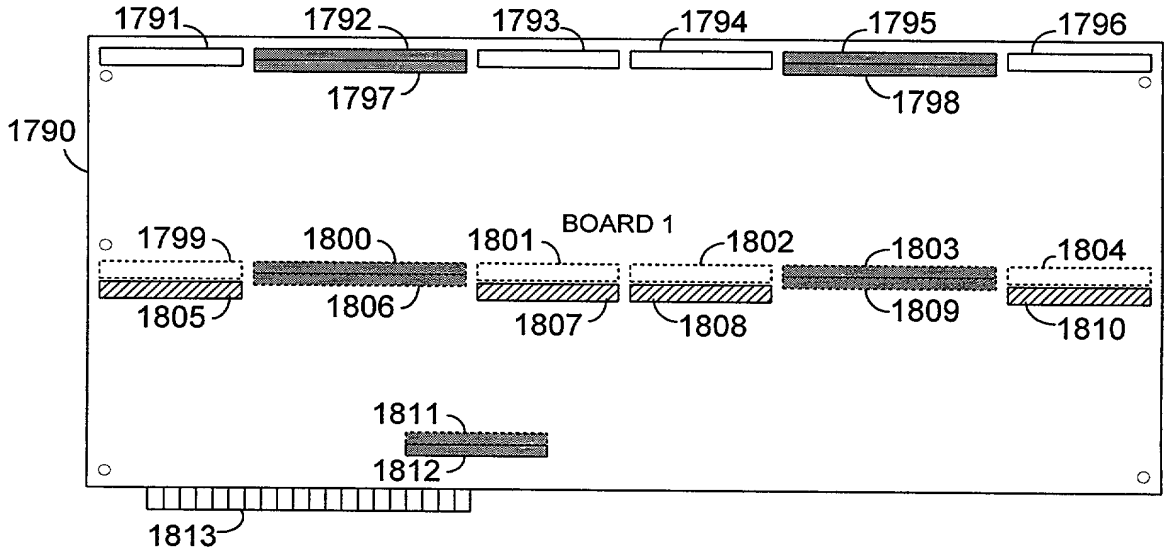


FIG. 41(F)

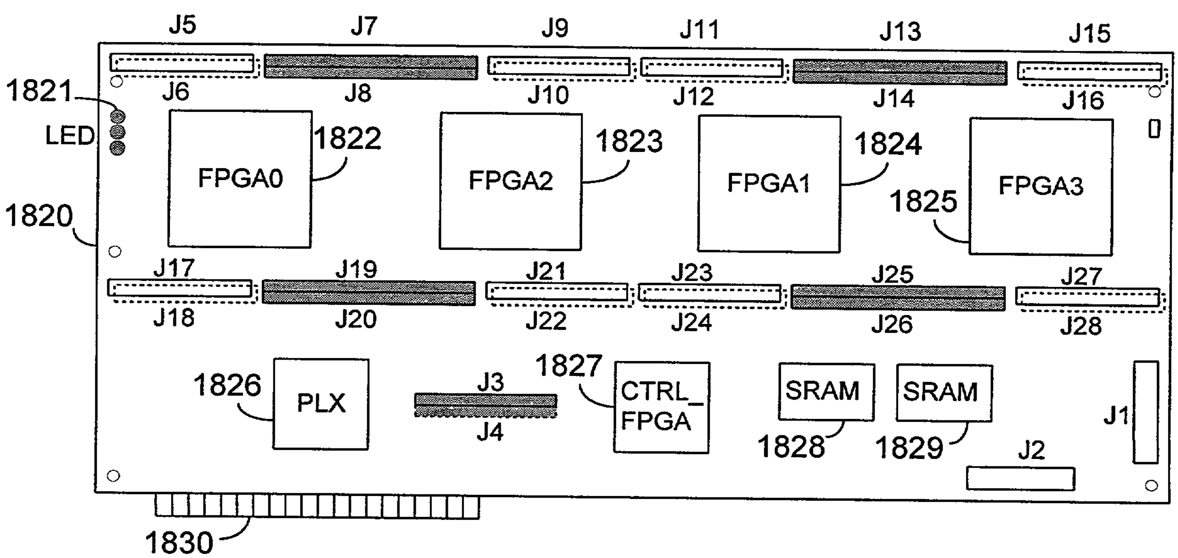

J5
J7
J9
J11
J13
J15
J6
J8
J10
J12
J14
J16
1821
LED
1820
FPGA0
1822
FPGA2
1823
FPGA1
1824
FPGA3
1825
J17
J19
J21
J23
J25
J27
J18
J20
J22
J24
J26
J28
1826
PLX
J3
J4
1827
CTRL_
FPGA
SRAM
SRAM
1828
1829
J1
J2
1830

FIG. 42

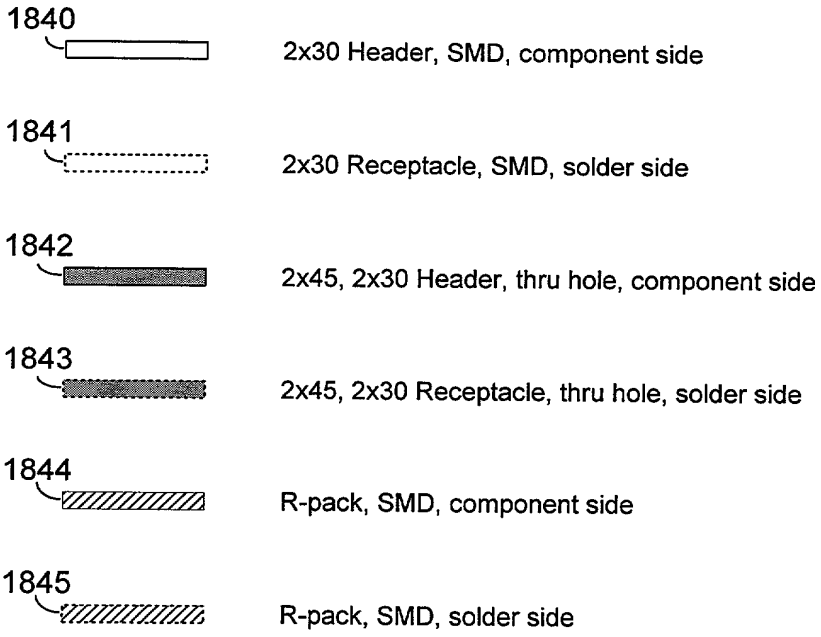

1840
2x30 Header, SMD, component side
1841
2x30 Receptacle, SMD, solder side
1842
2x45, 2x30 Header, thru hole, component side
1843
2x45, 2x30 Receptacle, thru hole, solder side
1844
R-pack, SMD, component side
1845
R-pack, SMD, solder side

FIG. 43

TWO-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

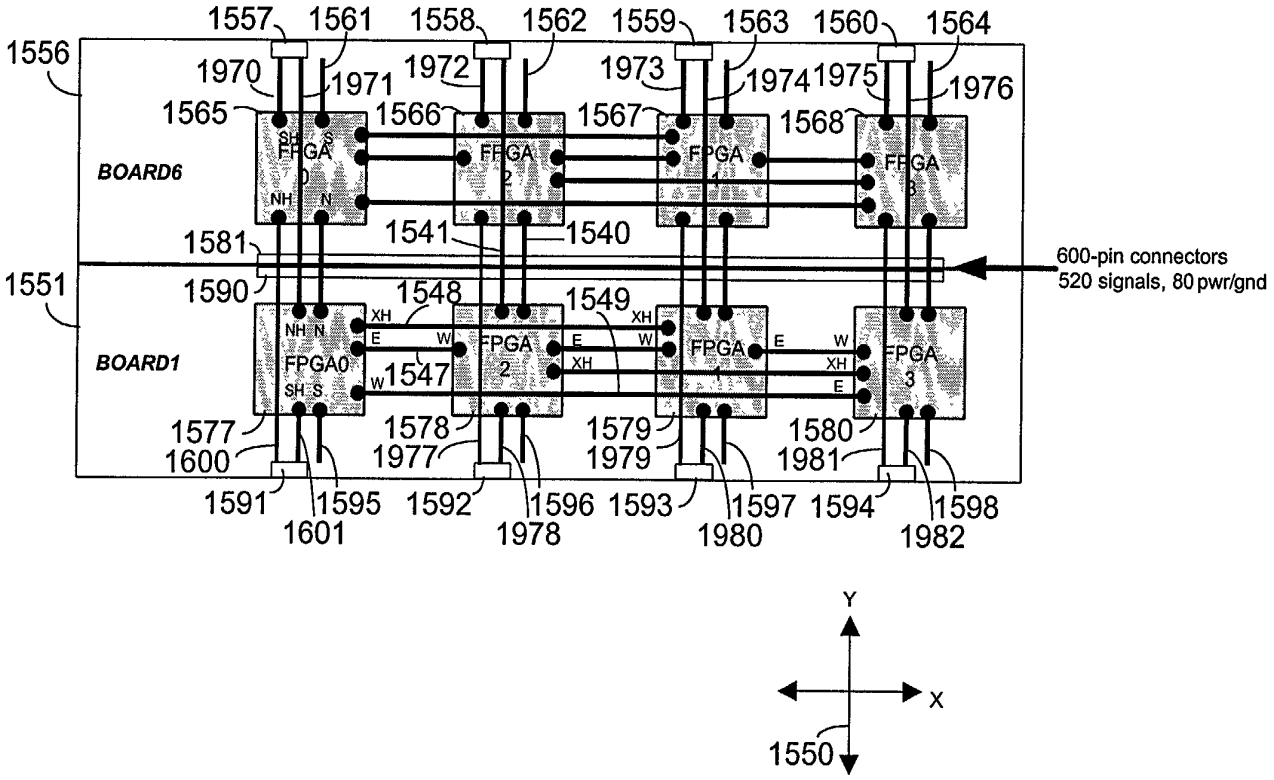


FIG. 44

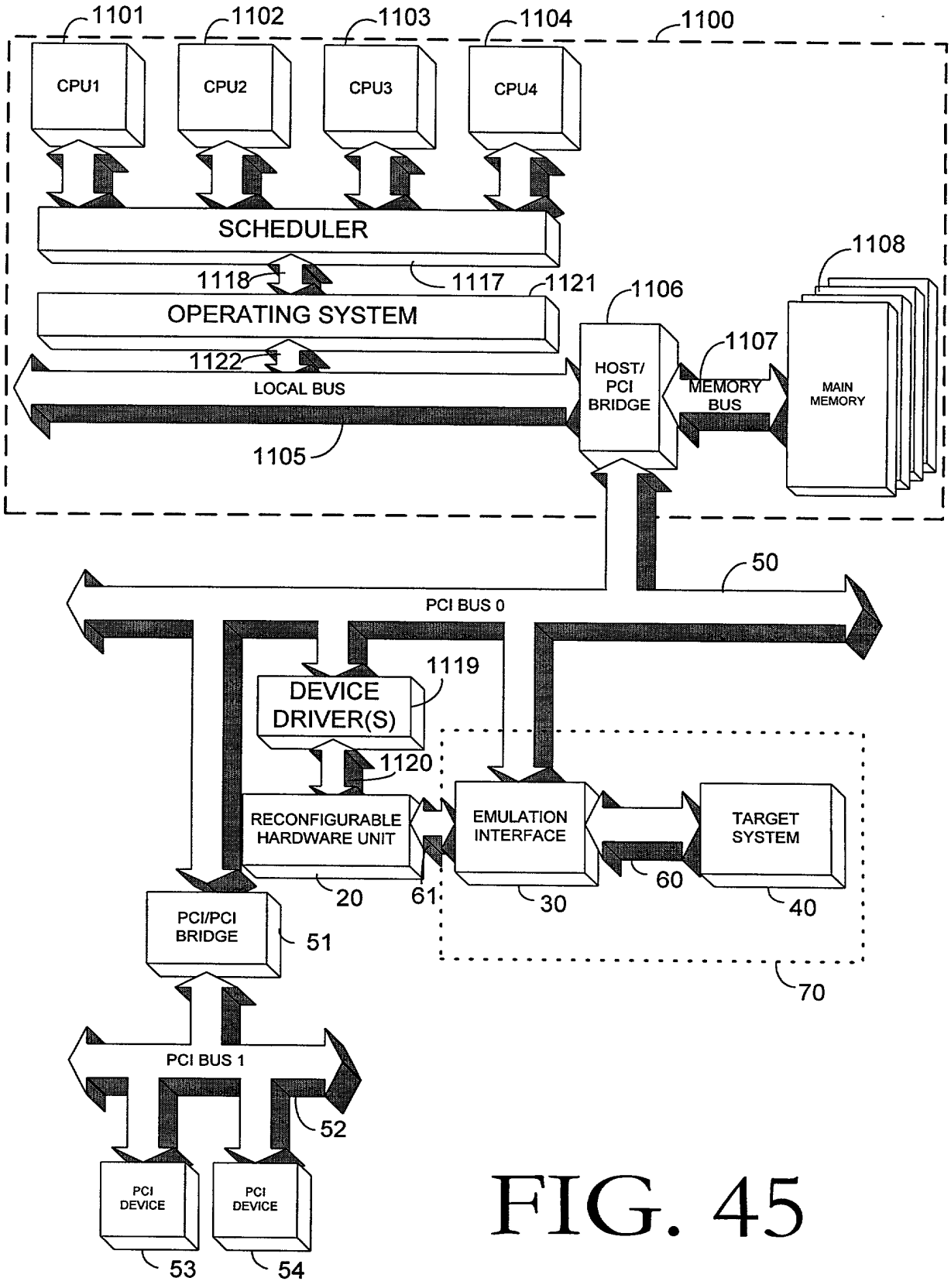

1101
1102
1103
1104
1100
CPU1
CPU2
CPU3
CPU4
SCHEDULER
1118
1117
1121
1106
1108
OPERATING SYSTEM
1122
1107
LOCAL BUS
HOST/
PCI
BRIDGE
MEMORY
BUS
MAIN
MEMORY
1105
50
PCI BUS 0
1119
DEVICE
DRIVER(S)
1120
RECONFIGURABLE
HARDWARE UNIT
EMULATION
INTERFACE
TARGET
SYSTEM
20
61
30
60
40
PCI/PCI
BRIDGE
51
70
PCI BUS 1
52
PCI
DEVICE
PCI
DEVICE
53
54

FIG. 45

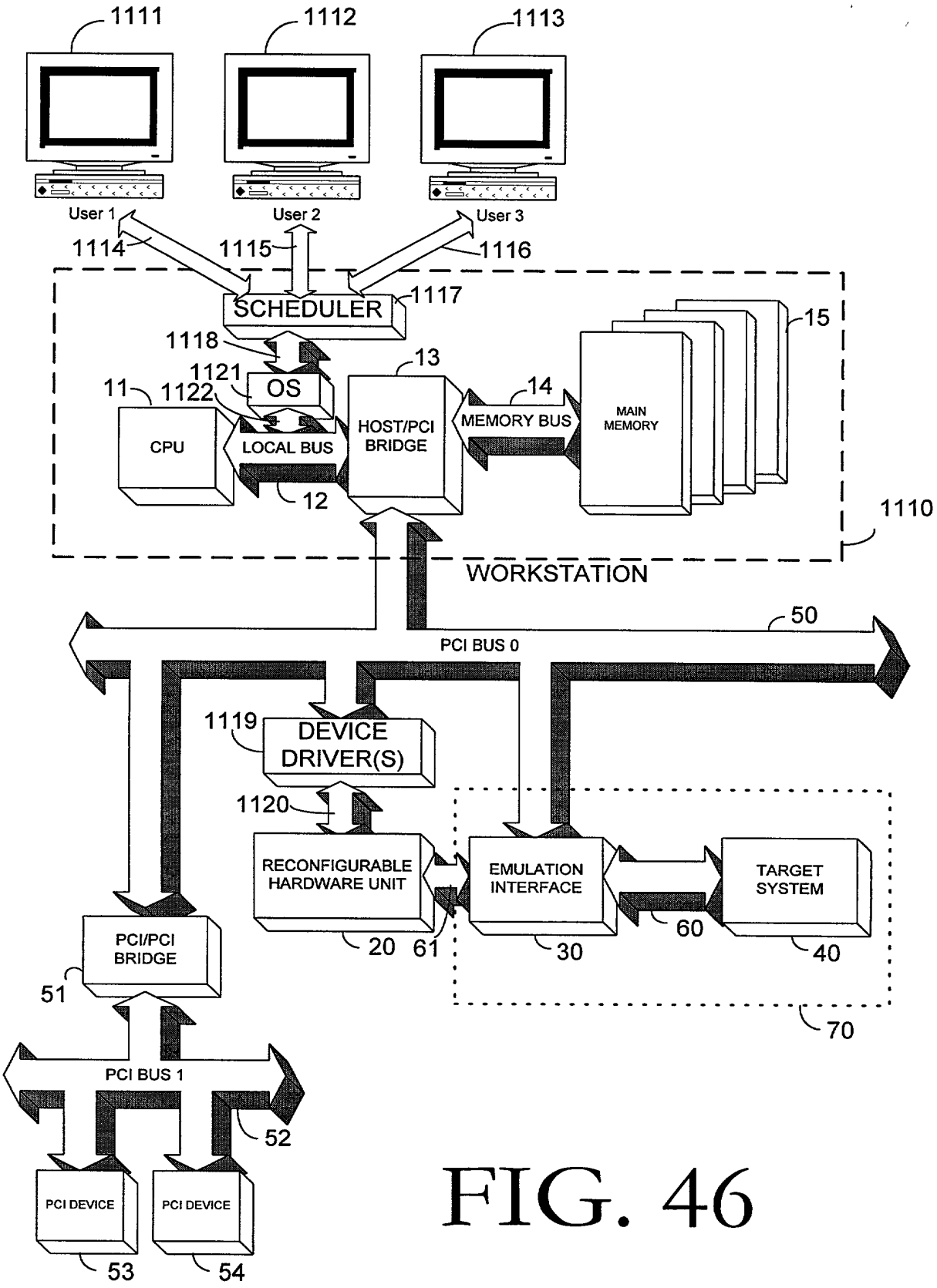

1111
1112
1113
User 1
User 2
User 3
1114
1115
1116
SCHEDULER
1117
1118
1121
OS
11
1122
CPU
LOCAL BUS
12
HOST/PCI BRIDGE
13
MEMORY BUS
14
MAIN MEMORY
15
1110
WORKSTATION
50
PCI BUS 0
1119
DEVICE DRIVER(S)
1120
RECONFIGURABLE HARDWARE UNIT
20
61
EMULATION INTERFACE
30
60
TARGET SYSTEM
40
70
PCI/PCI BRIDGE
51
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 46

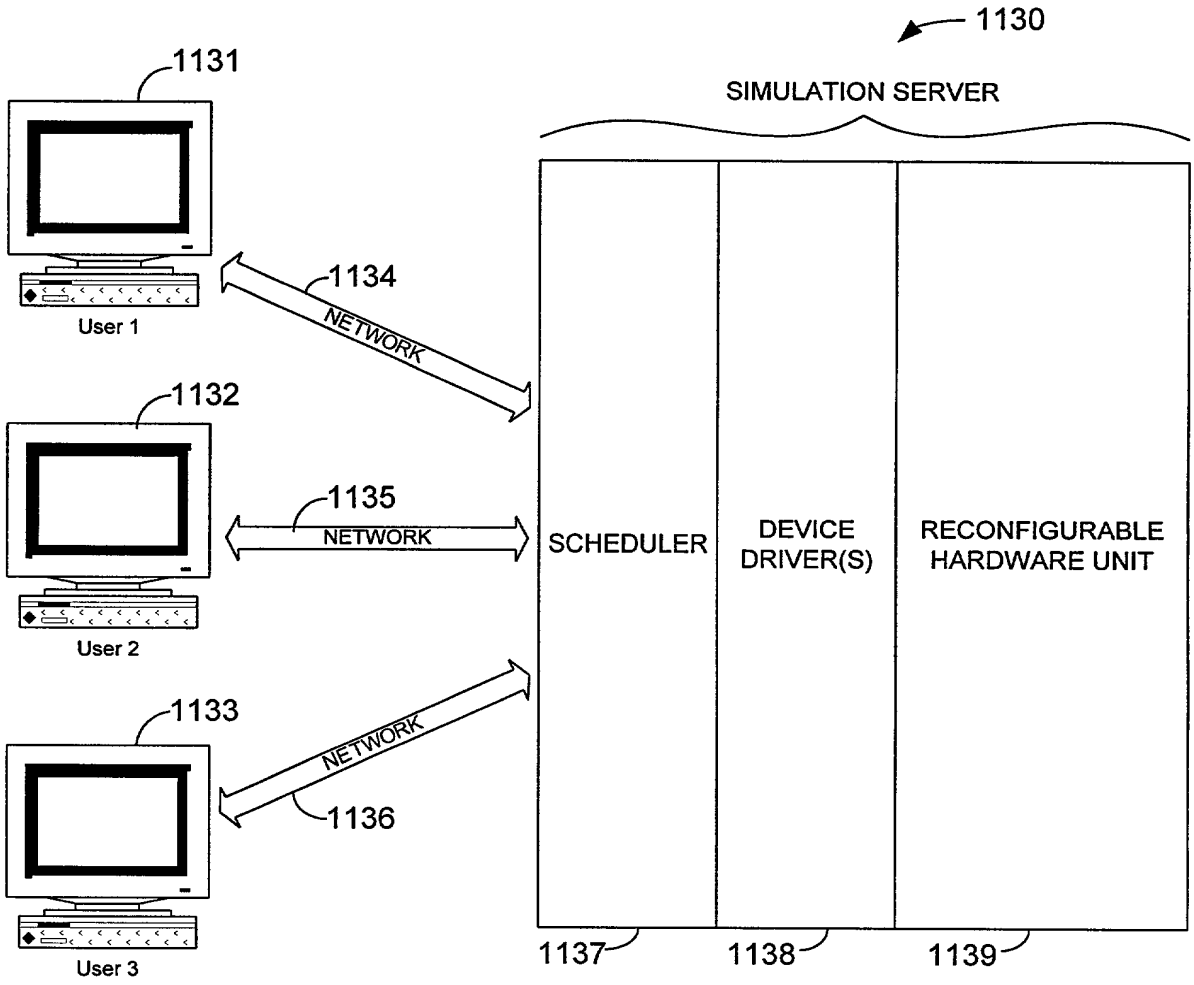

1130
SIMULATION SERVER
1131
User 1
1134
NETWORK
1132
User 2
1135
NETWORK
1133
User 3
NETWORK
1136
SCHEDULER
DEVICE DRIVER(S)
RECONFIGURABLE HARDWARE UNIT
1137
1138
1139

FIG. 47

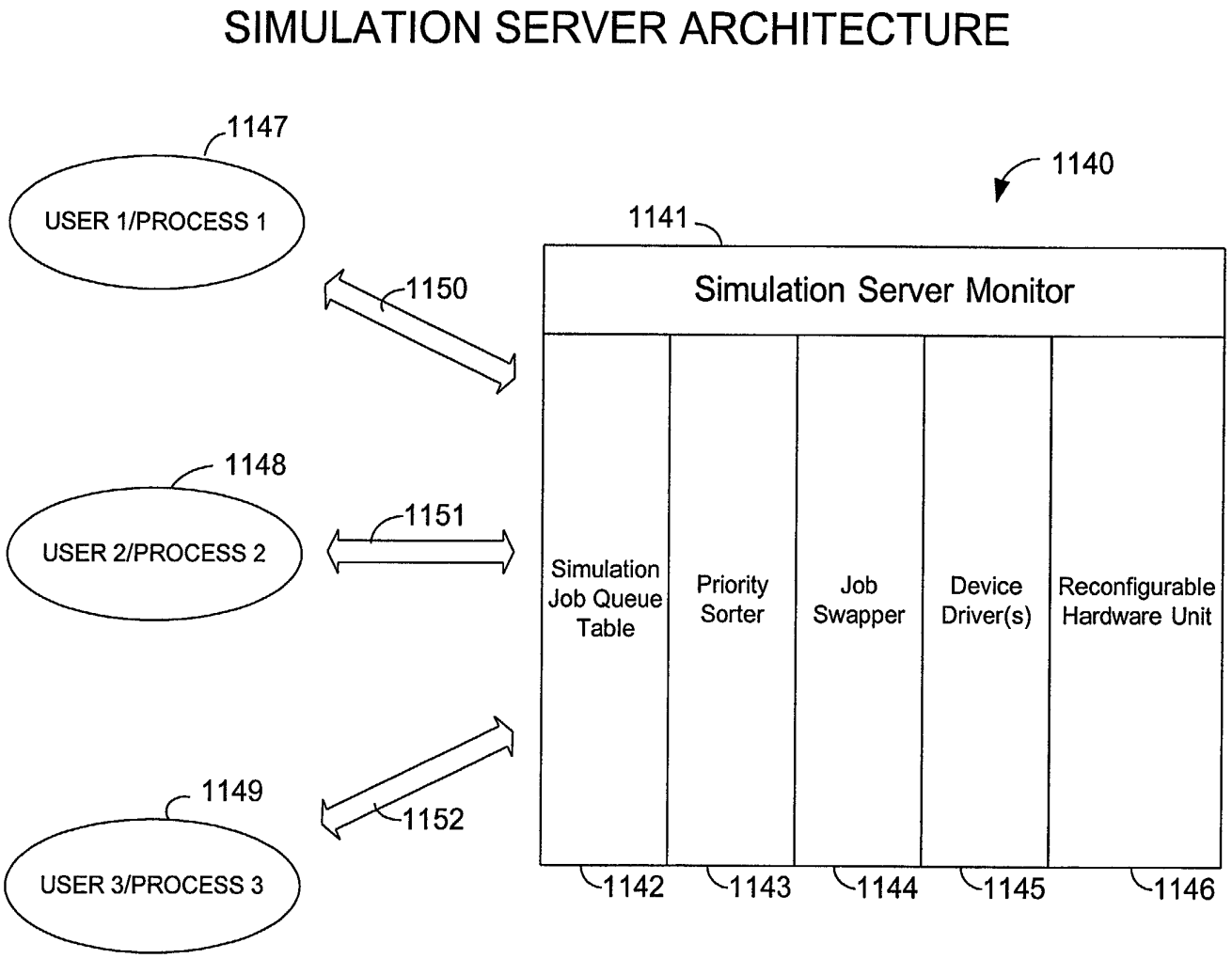

SIMULATION SERVER ARCHITECTURE
1147
USER 1/PROCESS 1
1140
1141
Simulation Server Monitor
1150
1148
USER 2/PROCESS 2
1151
Simulation Job Queue Table
Priority Sorter
Job Swapper
Device Driver(s)
Reconfigurable Hardware Unit
1149
1152
USER 3/PROCESS 3
1142
1143
1144
1145
1146

FIG. 48

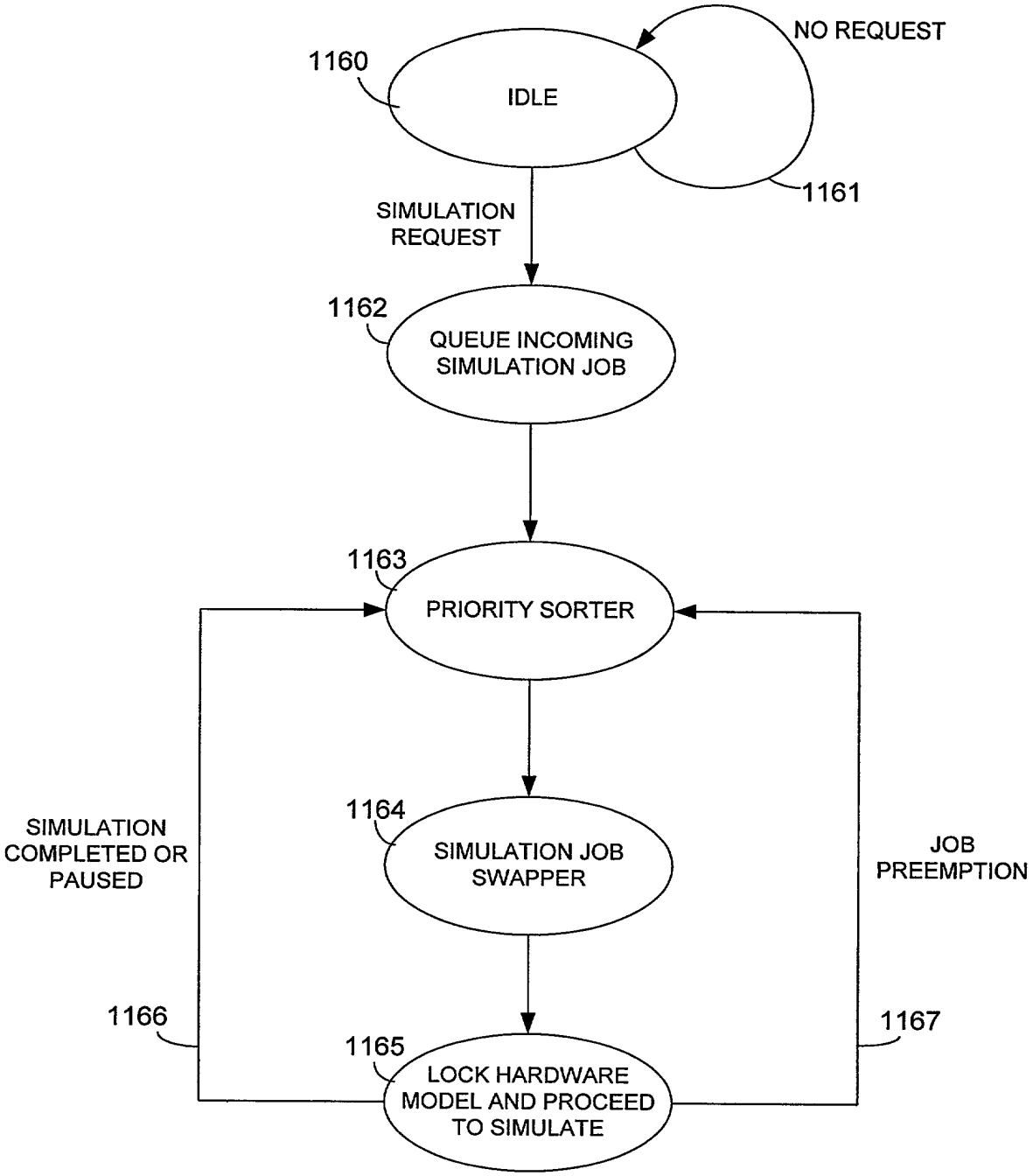

NO REQUEST
1160
IDLE
1161
SIMULATION REQUEST
1162
QUEUE INCOMING SIMULATION JOB
1163
PRIORITY SORTER
1164
SIMULATION JOB SWAPPER
SIMULATION COMPLETED OR PAUSED
JOB PREEMPTION
1166
1167
1165
LOCK HARDWARE MODEL AND PROCEED TO SIMULATE

FIG. 49

JOB SWAPPER

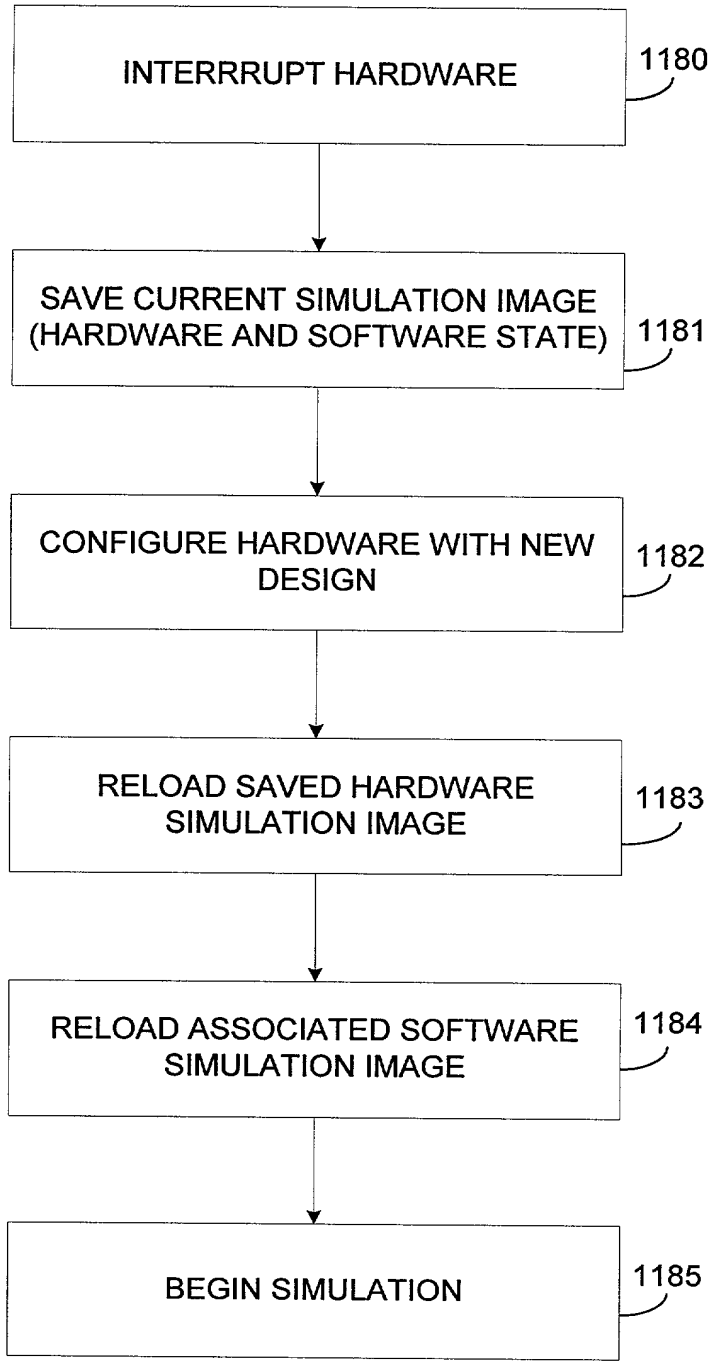


FIG. 50

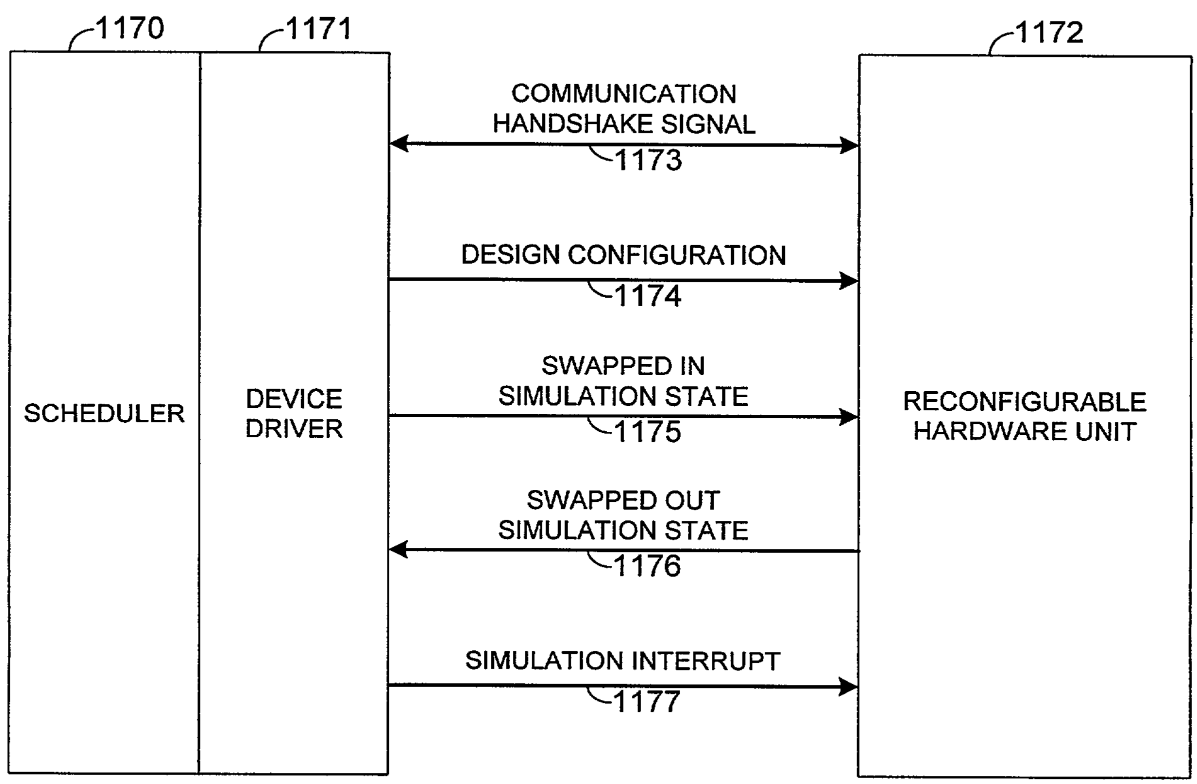

1170
1171
1172
SCHEDULER
DEVICE DRIVER
RECONFIGURABLE HARDWARE UNIT
COMMUNICATION HANDSHAKE SIGNAL
1173
DESIGN CONFIGURATION
1174
SWAPPED IN SIMULATION STATE
1175
SWAPPED OUT SIMULATION STATE
1176
SIMULATION INTERRUPT
1177

FIG. 51

PRIORITY I { JOB A, JOB B }

PRIORITY II { JOB C, JOB D }

TIME-SHARED HARDWARE USAGE:

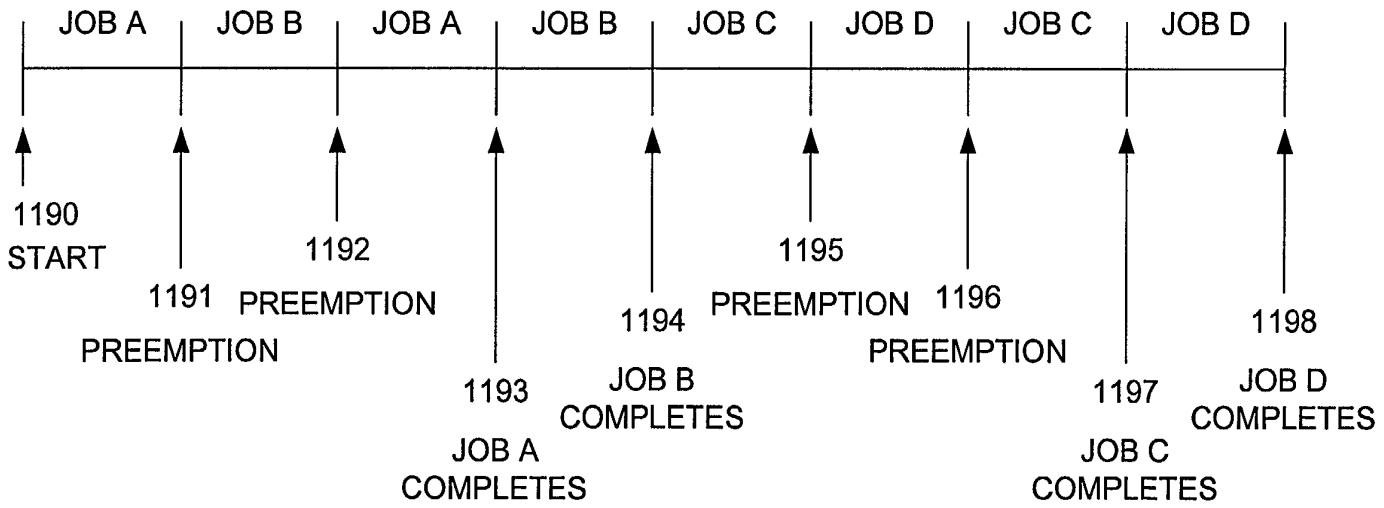


FIG. 52

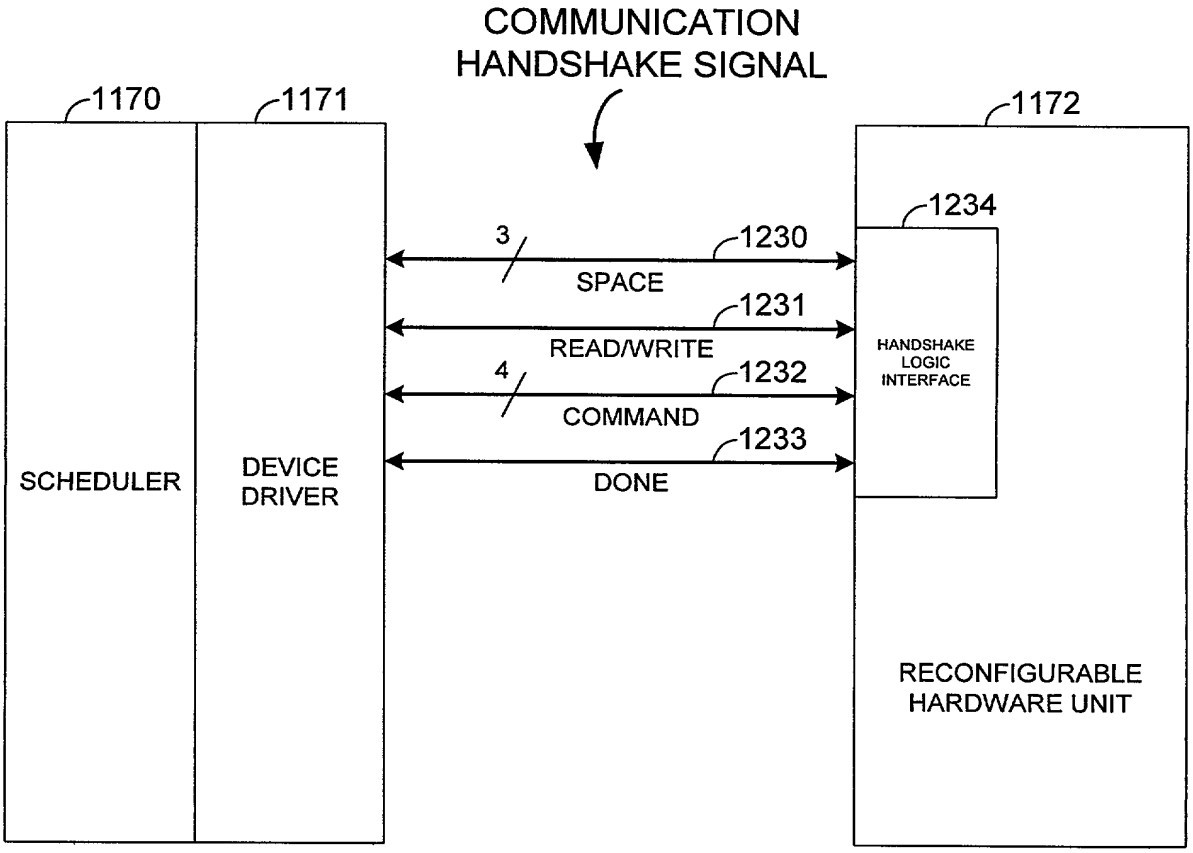

COMMUNICATION
HANDSHAKE SIGNAL
1170
1171
1172
1234
3
1230
SPACE
1231
READ/WRITE
HANDSHAKE
LOGIC
INTERFACE
4
1232
COMMAND
1233
DONE
SCHEDULER
DEVICE
DRIVER
RECONFIGURABLE
HARDWARE UNIT

FIG. 53

COMMUNICATION HANDSHAKE PROTOCOL

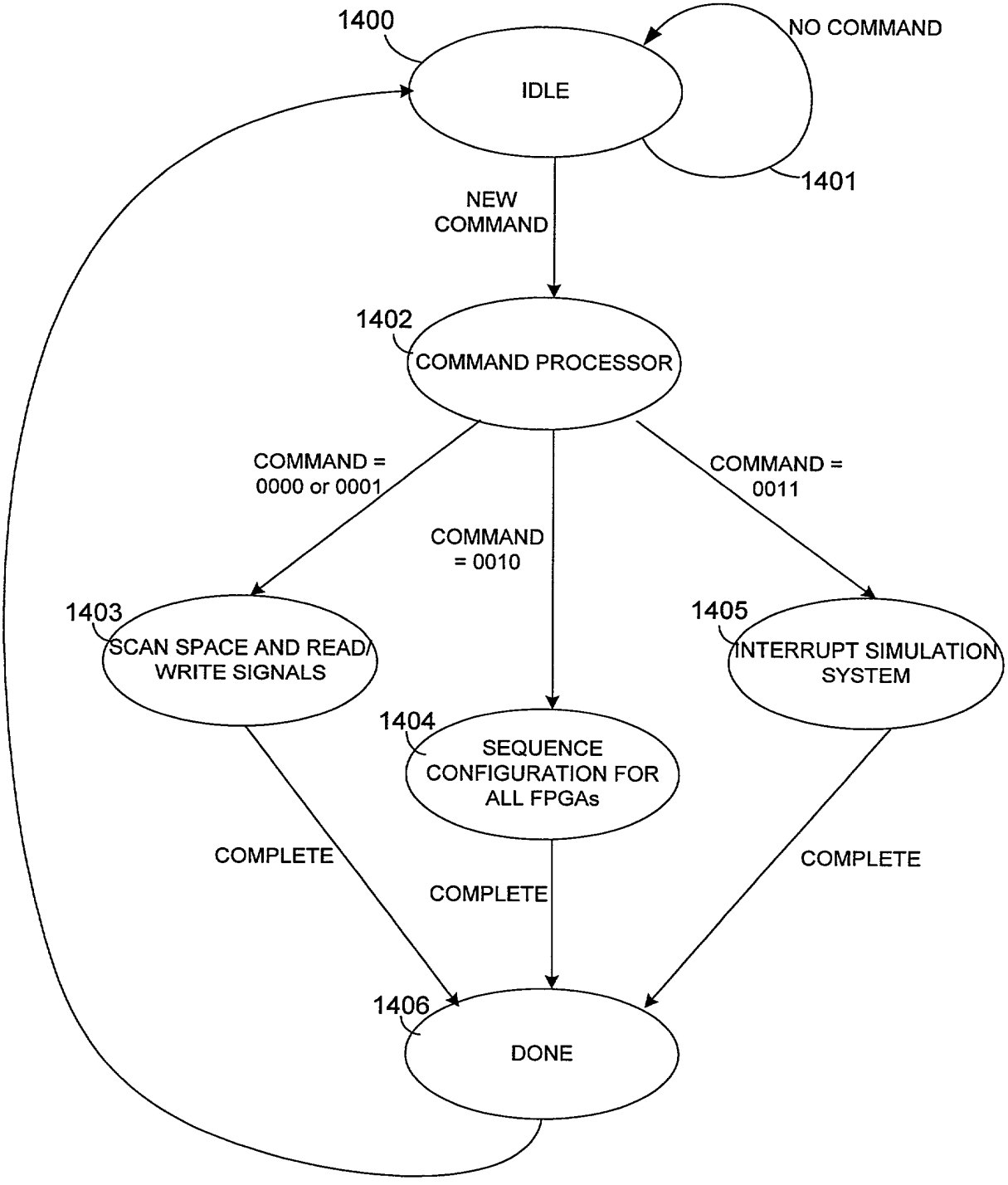


FIG. 54

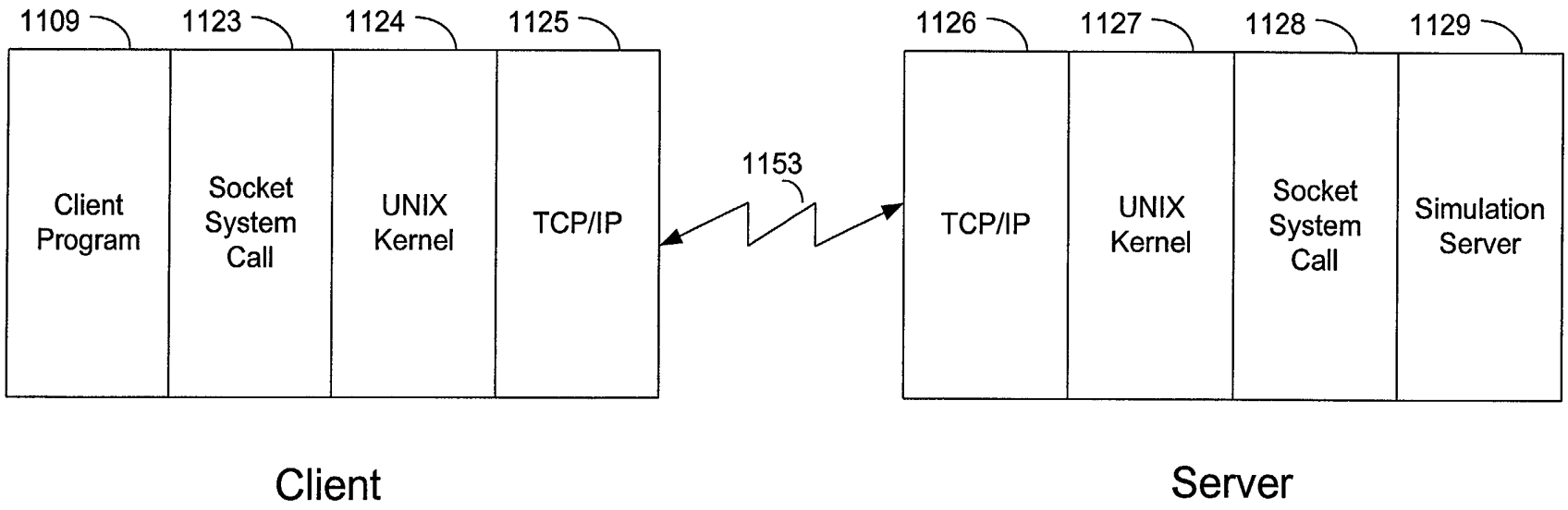


FIG. 55

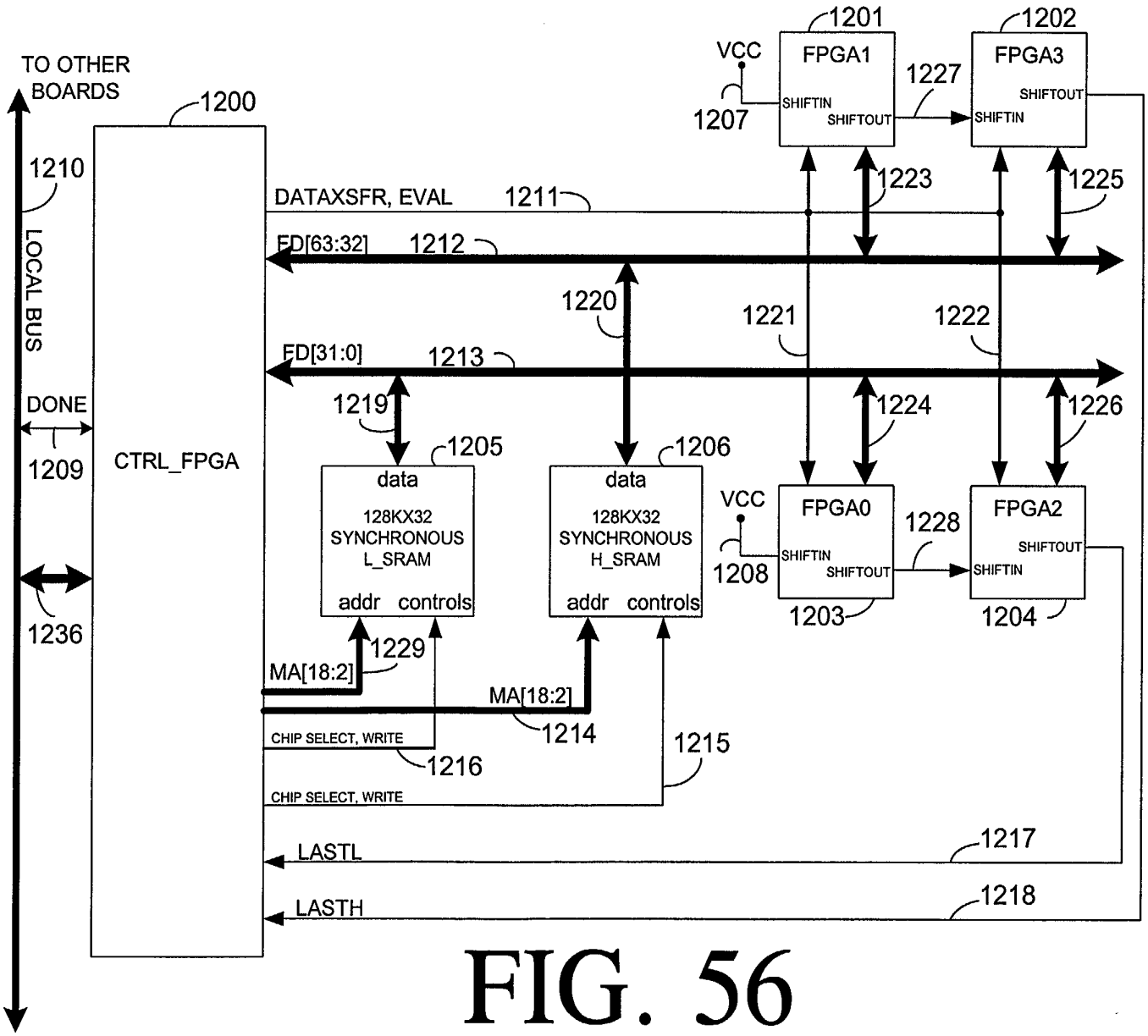


FIG. 56

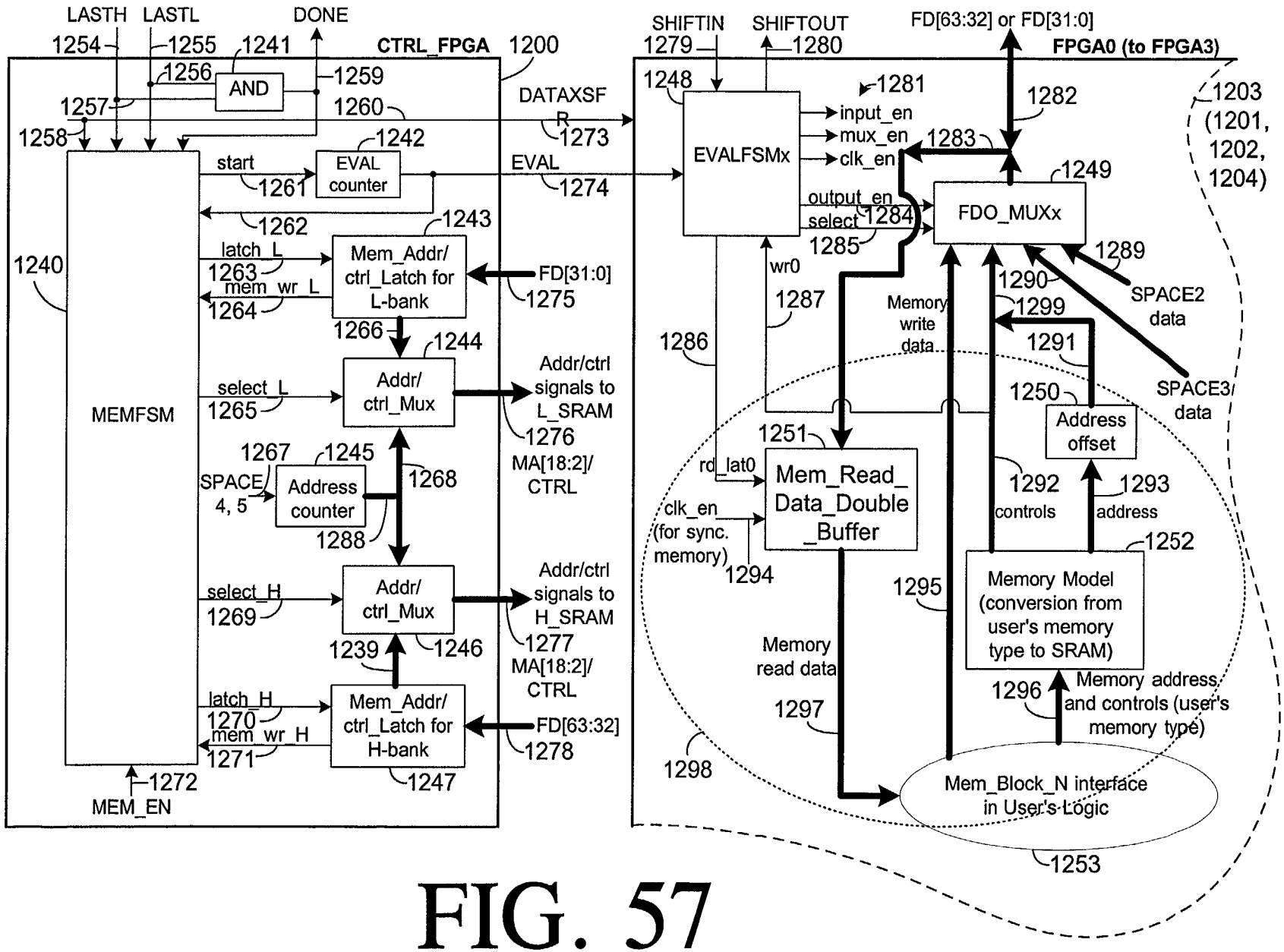

LASTH
LASTL
DONE
1254
1255
1241
CTRL_FPGA
1200
1256
AND
1259
1257
1260
1258
1242
start
EVAL counter
1261
DATAXSFR
1273
EVAL
1274
1262
1243
latch_L
1263
mem_wr_L
1264
Mem_Addr/ ctrl_Latch for L-bank
FD[31:0]
1275
1240
1266
1244
Addr/ ctrl_Mux
select_L
1265
Addr/ctrl signals to L_SRAM
1276
MA[18:2]/ CTRL
MEMFSM
1267
1245
SPACE 4, 5
Address counter
1268
1288
select_H
1269
Addr/ ctrl_Mux
Addr/ctrl signals to H_SRAM
1277
MA[18:2]/ CTRL
1239
1246
latch_H
1270
mem_wr_H
1271
Mem_Addr/ ctrl_Latch for H-bank
FD[63:32]
1278
1247
1272
MEM_EN
SHIFTIN
1279
SHIFTOUT
1280
FD[63:32] or FD[31:0]
FPGA0 (to FPGA3)
1248
1281
1282
1203 (1201, 1202, 1204)
EVALFSMx
input_en
mux_en
clk_en
1283
1249
output_en
select
1284
FDO_MUXx
1285
1289
wr0
1290
SPACE2 data
1287
Memory write data
1299
1286
1291
SPACE3 data
1250
Address offset
1251
rd_lat0
Mem_Read_ Data_Double _Buffer
1292
controls
1293
address
clk_en (for sync. memory)
1294
1252
1295
Memory Model (conversion from user's memory type to SRAM)
Memory read data
1296
Memory address and controls (user's memory type)
1297
1298
Mem_Block_N interface in User's Logic
1253

FIG. 57

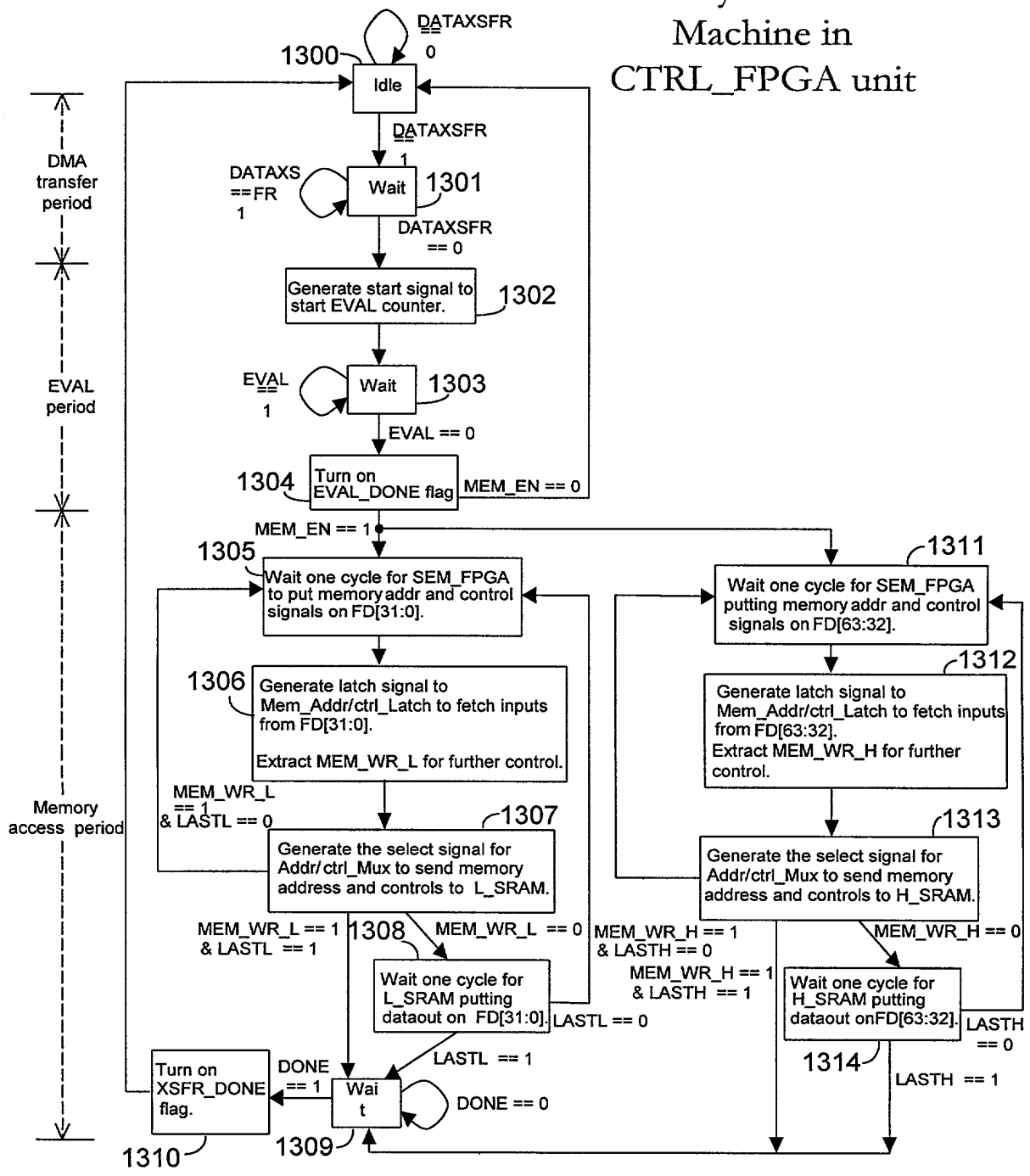

MEMFSM - Memory Finite State Machine in CTRL_FPGA unit
DMA transfer period
EVAL period
Memory access period
1300
Idle
DATAXSFR == 0
DATAXSFR == 1
DATAXS ==FR 1
Wait
1301
DATAXSFR == 0
Generate start signal to start EVAL counter.
1302
EVAL == 1
Wait
1303
EVAL == 0
1304
Turn on EVAL_DONE flag
MEM_EN == 0
MEM_EN == 1
1305
Wait one cycle for SEM_FPGA to put memory addr and control signals on FD[31:0].
1306
Generate latch signal to Mem_Addr/ctrl_Latch to fetch inputs from FD[31:0].
Extract MEM_WR_L for further control.
MEM_WR_L == 1 & LASTL == 0
1307
Generate the select signal for Addr/ctrl_Mux to send memory address and controls to L_SRAM.
MEM_WR_L == 1 & LASTL == 1
1308
MEM_WR_L == 0
Wait one cycle for L_SRAM putting dataout on FD[31:0].
LASTL == 0
LASTL == 1
Turn on XSFR_DONE flag.
DONE == 1
Wai t
DONE == 0
1310
1309
1311
Wait one cycle for SEM_FPGA putting memory addr and control signals on FD[63:32].
1312
Generate latch signal to Mem_Addr/ctrl_Latch to fetch inputs from FD[63:32].
Extract MEM_WR_H for further control.
1313
Generate the select signal for Addr/ctrl_Mux to send memory address and controls to H_SRAM.
MEM_WR_H == 1 & LASTH == 0
MEM_WR_H == 1 & LASTH == 1
MEM_WR_H == 0
Wait one cycle for H_SRAM putting dataout onFD[63:32].
LASTH == 0
1314
LASTH == 1

FIG. 58

EVALFSM - EVAL
Finite State Machine in
each FPGA logic device

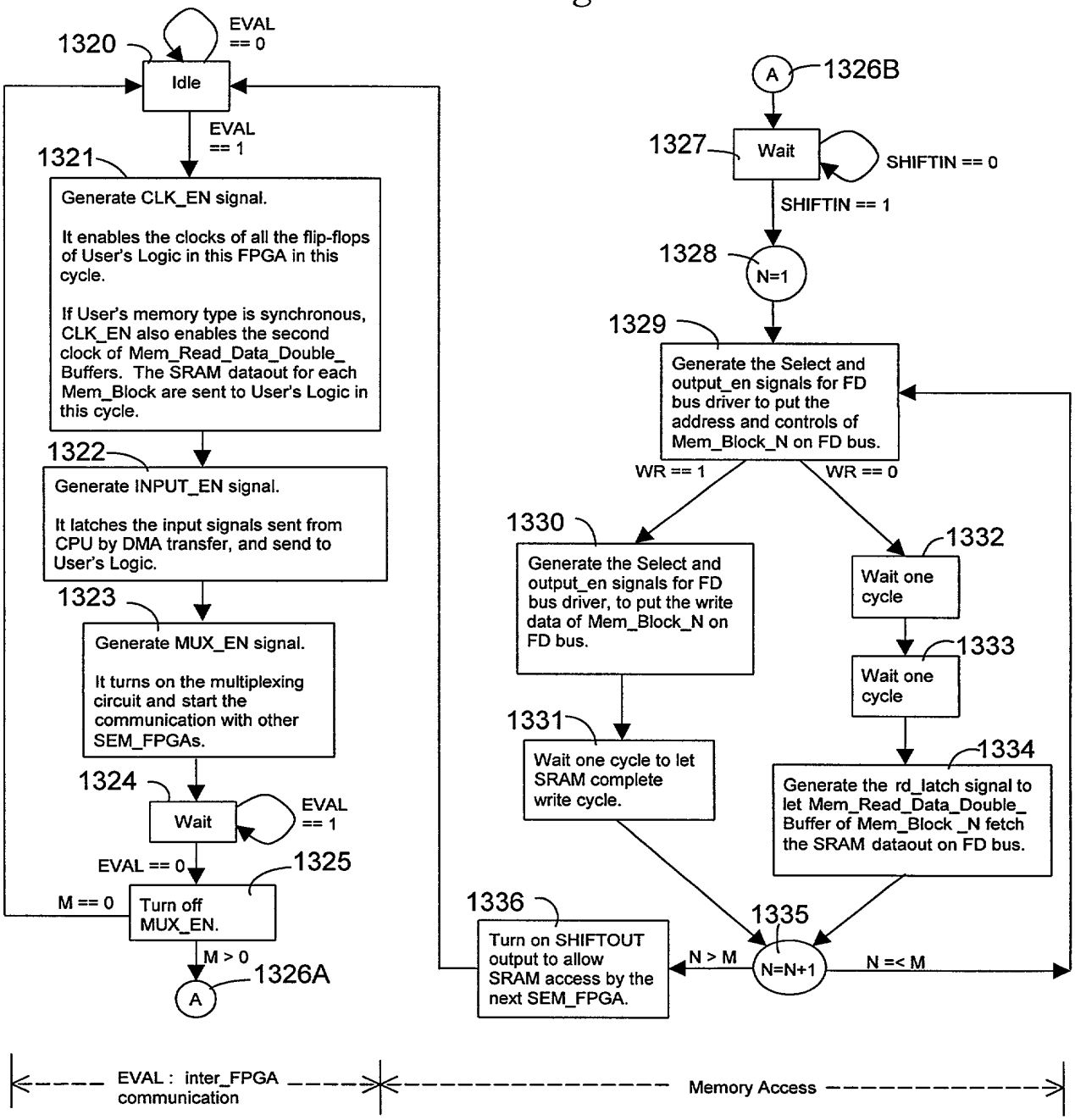


FIG. 59

MEMORY READ DATA DOUBLE BUFFER

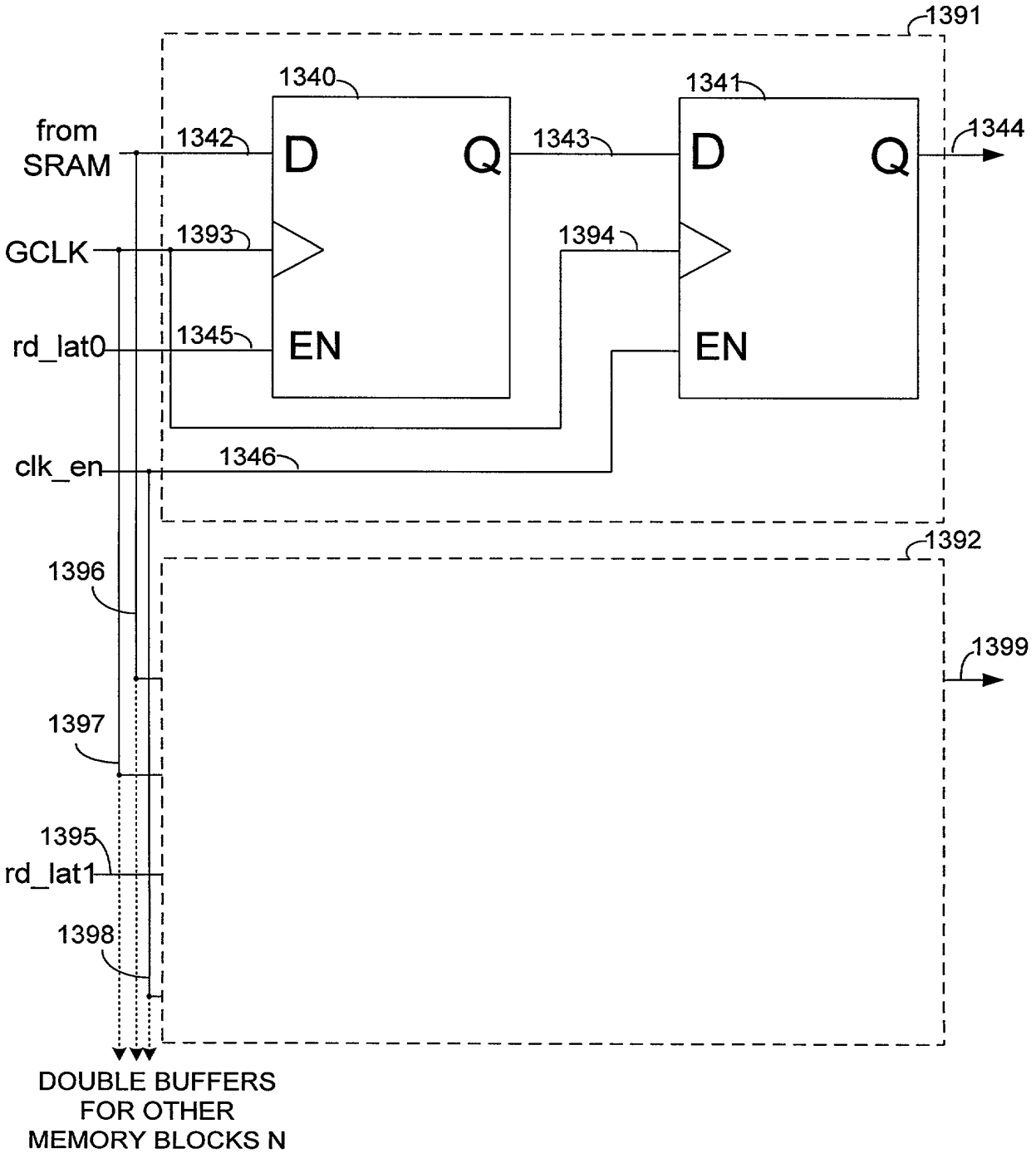


FIG. 60

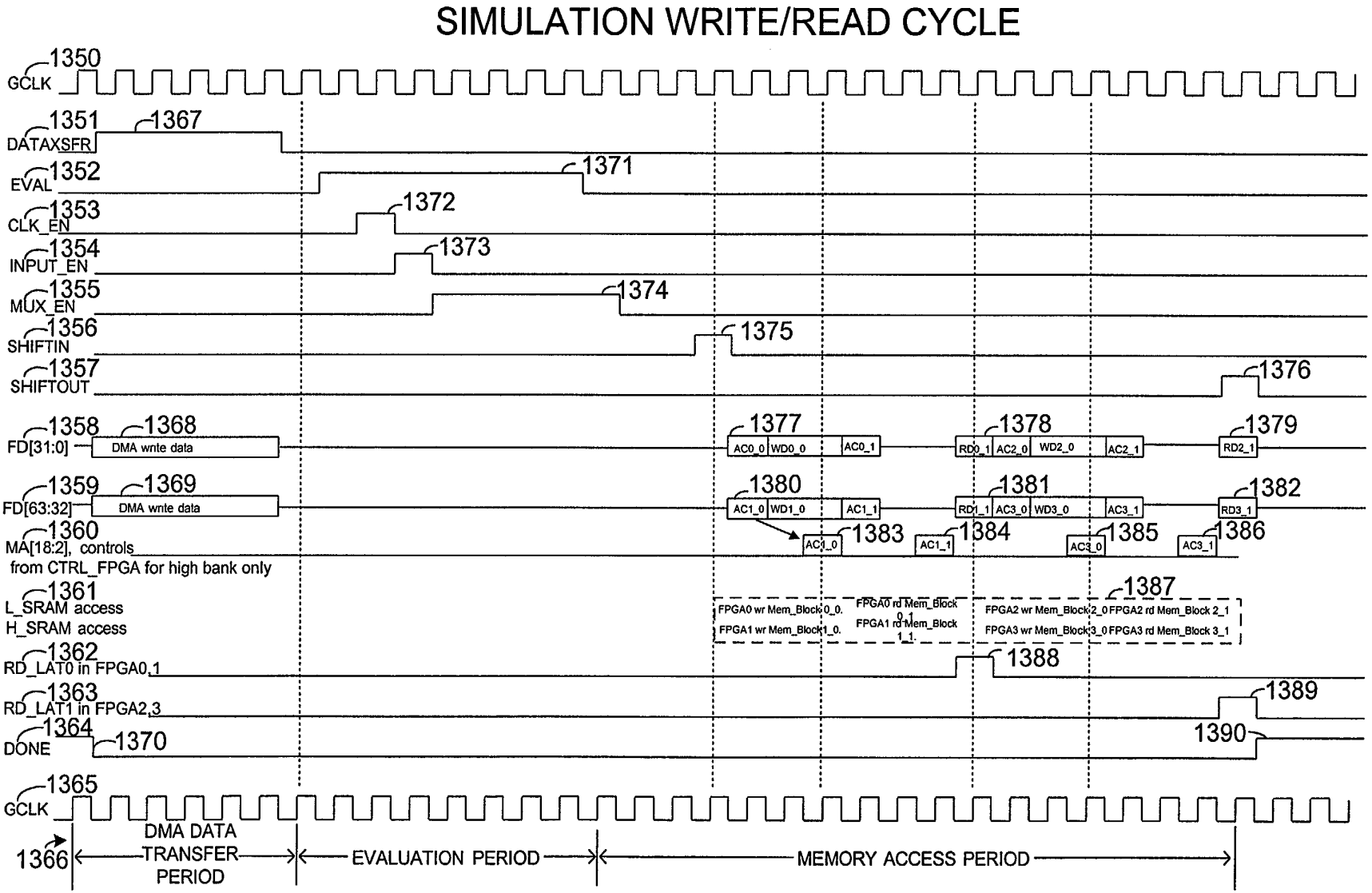

SIMULATION WRITE/READ CYCLE
1350
GCLK
1351
DATAXSFR
1367
1352
EVAL
1371
1353
CLK_EN
1372
1354
INPUT_EN
1373
1355
MUX_EN
1374
1356
SHIFTIN
1375
1357
SHIFTOUT
1376
1358
FD[31:0]
1368
DMA wnte data
1377
AC0_0
WD0_0
AC0_1
1378
RD0_1
AC2_0
WD2_0
AC2_1
1379
RD2_1
1359
FD[63:32]
1369
DMA wnte data
1380
AC1_0
WD1_0
AC1_1
1381
RD1_1
AC3_0
WD3_0
AC3_1
1382
RD3_1
1360
MA[18:2], controls
from CTRL_FPGA for high bank only
1383
1384
1385
1386
1361
L_SRAM access
H_SRAM access
1387
FPGA0 wr Mem_Block 0_0
FPGA0 rd Mem_Block 0_1
FPGA1 wr Mem_Block 1_0
FPGA1 rd Mem_Block 1_1
FPGA2 wr Mem_Block 2_0
FPGA2 rd Mem_Block 2_1
FPGA3 wr Mem_Block 3_0
FPGA3 rd Mem_Block 3_1
1362
RD_LAT0 in FPGA0,1
1388
1363
RD_LAT1 in FPGA2,3
1389
1364
DONE
1370
1390
1365
GCLK
1366
DMA DATA TRANSFER PERIOD
EVALUATION PERIOD
MEMORY ACCESS PERIOD

FIG. 61

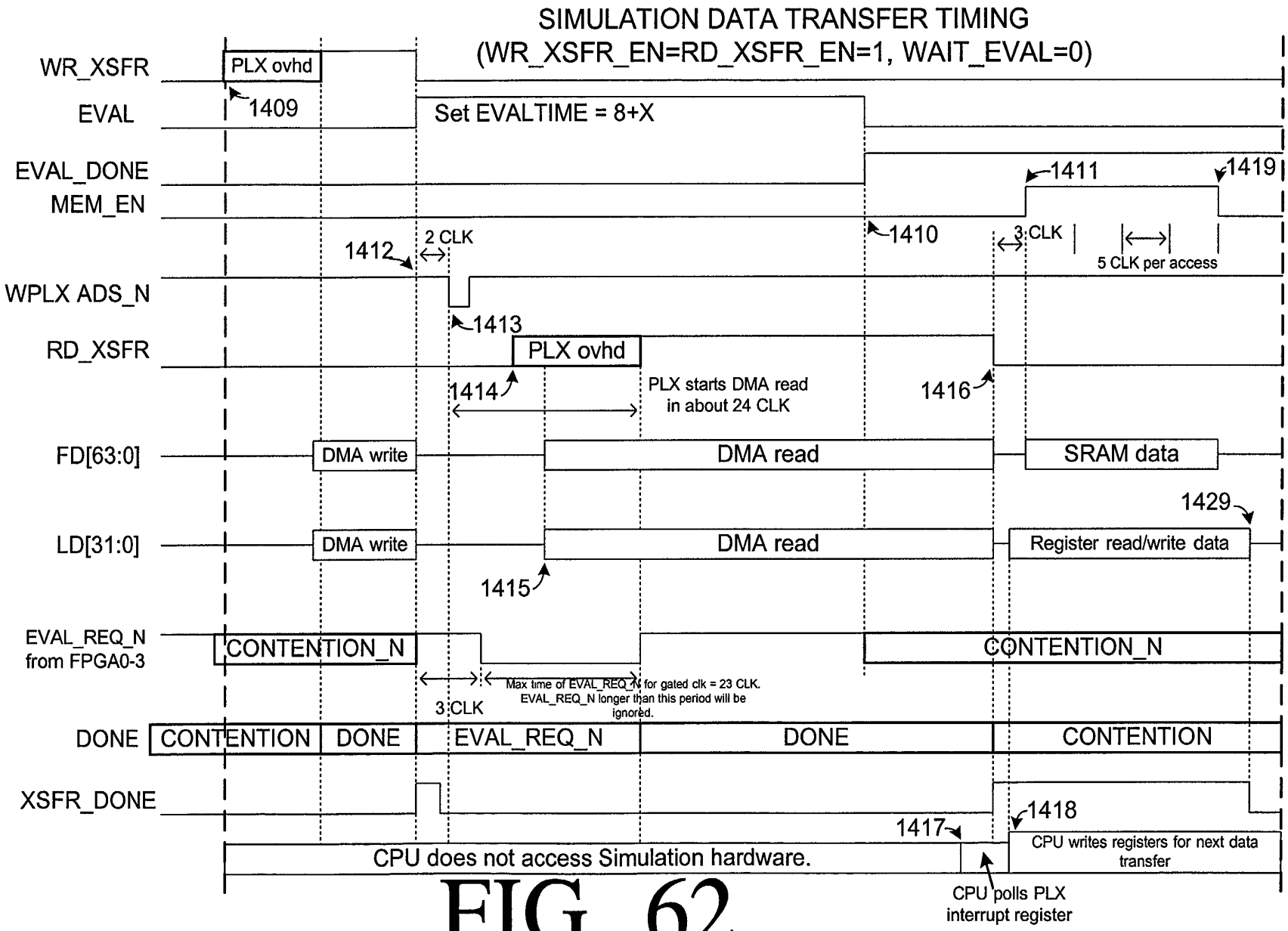

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=0)
WR_XSFR
PLX ovhd
1409
EVAL
Set EVALTIME = 8+X
EVAL_DONE
MEM_EN
1411
1419
1410
3 CLK
5 CLK per access
1412
2 CLK
WPLX ADS_N
1413
RD_XSFR
PLX ovhd
1414
PLX starts DMA read
in about 24 CLK
1416
FD[63:0]
DMA write
DMA read
SRAM data
1429
LD[31:0]
DMA write
DMA read
Register read/write data
1415
EVAL_REQ_N
from FPGA0-3
CONTENTION_N
CONTENTION_N
Max time of EVAL_REQ_N for gated clk = 23 CLK.
EVAL_REQ_N longer than this period will be
ignored.
3 CLK
DONE
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
XSFR_DONE
1418
1417
CPU writes registers for next data
transfer
CPU does not access Simulation hardware.
CPU polls PLX
interrupt register

FIG. 62

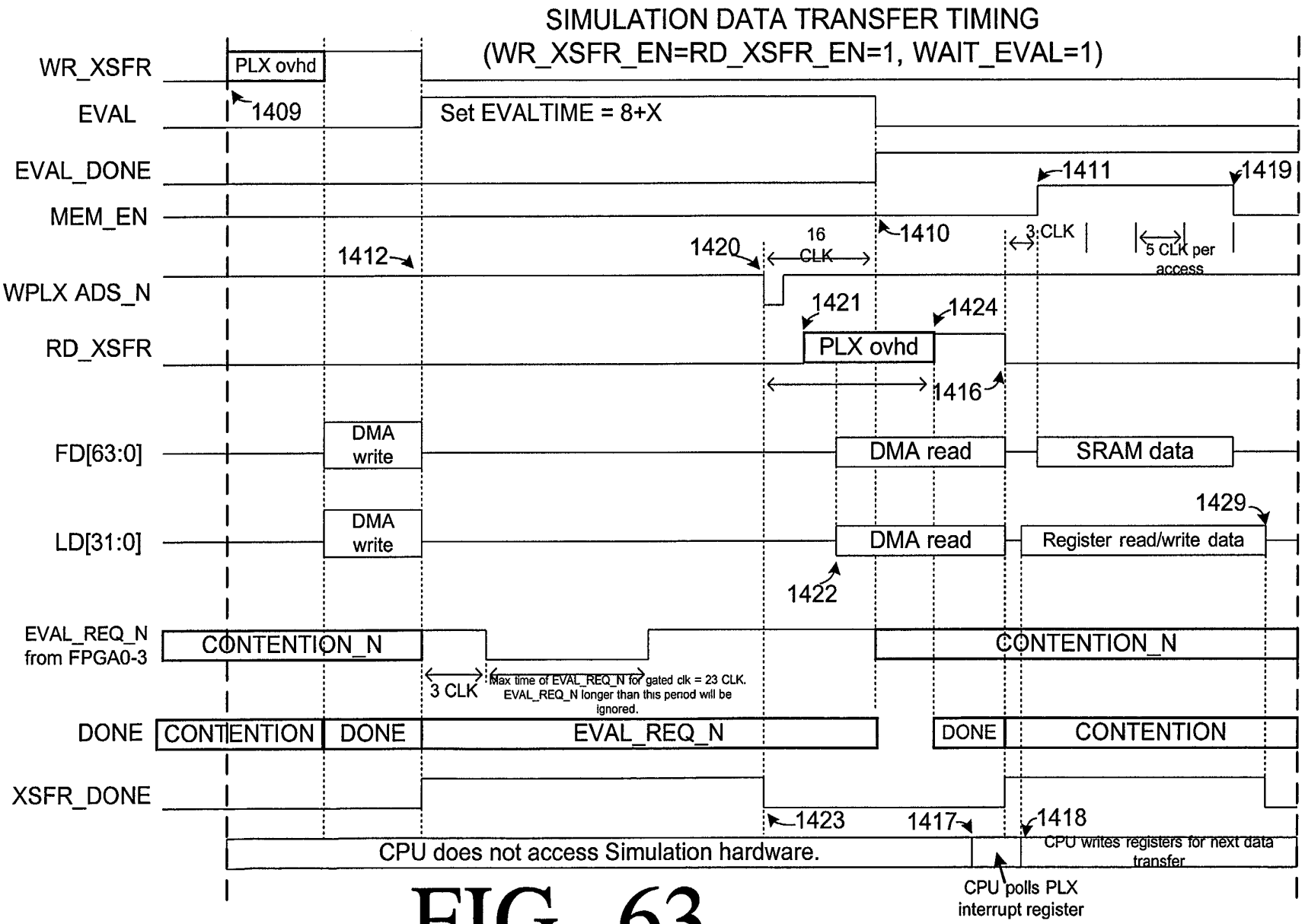

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=1)
WR_XSFR
EVAL
EVAL_DONE
MEM_EN
WPLX ADS_N
RD_XSFR
FD[63:0]
LD[31:0]
EVAL_REQ_N from FPGA0-3
DONE
XSFR_DONE
PLX ovhd
1409
Set EVALTIME = 8+X
1411
1419
1410
3 CLK
5 CLK per access
1412
1420
16 CLK
1421
1424
PLX ovhd
1416
DMA write
DMA read
SRAM data
1429
DMA write
DMA read
Register read/write data
1422
CONTENTION_N
CONTENTION_N
3 CLK
Max time of EVAL_REQ_N for gated clk = 23 CLK. EVAL_REQ_N longer than this period will be ignored.
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
1423
1417
1418
CPU writes registers for next data transfer
CPU does not access Simulation hardware.
CPU polls PLX interrupt register

FIG. 63

Typical User Design of PCI Add-on Cards

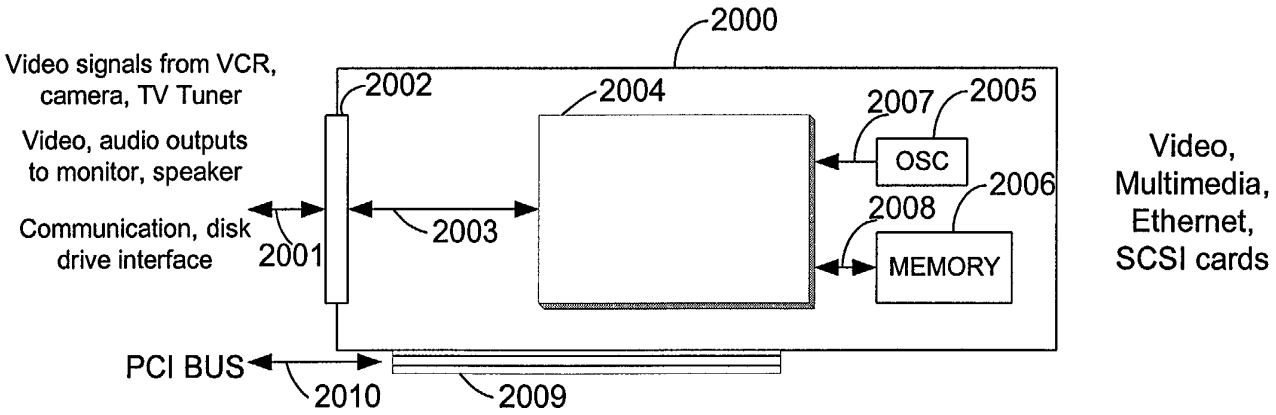
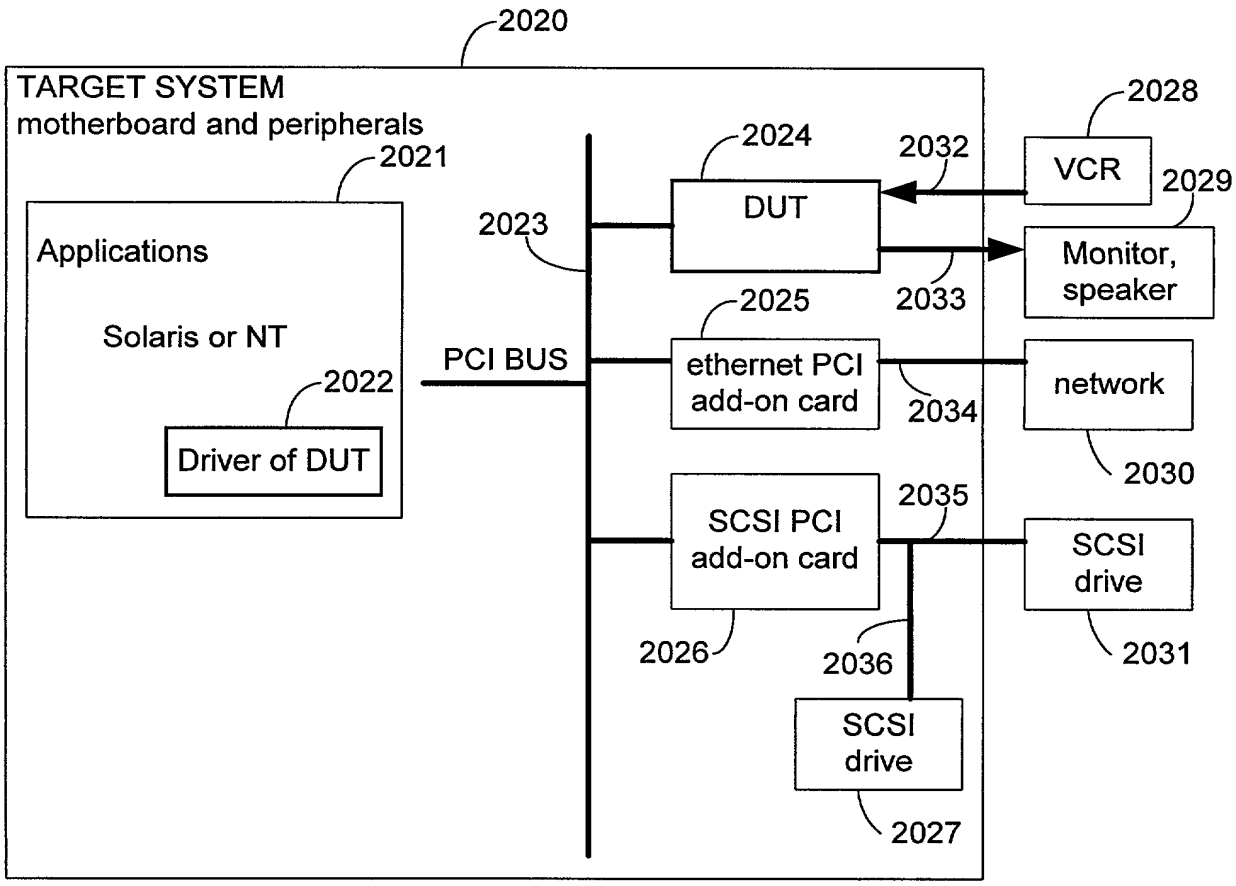


FIG. 64

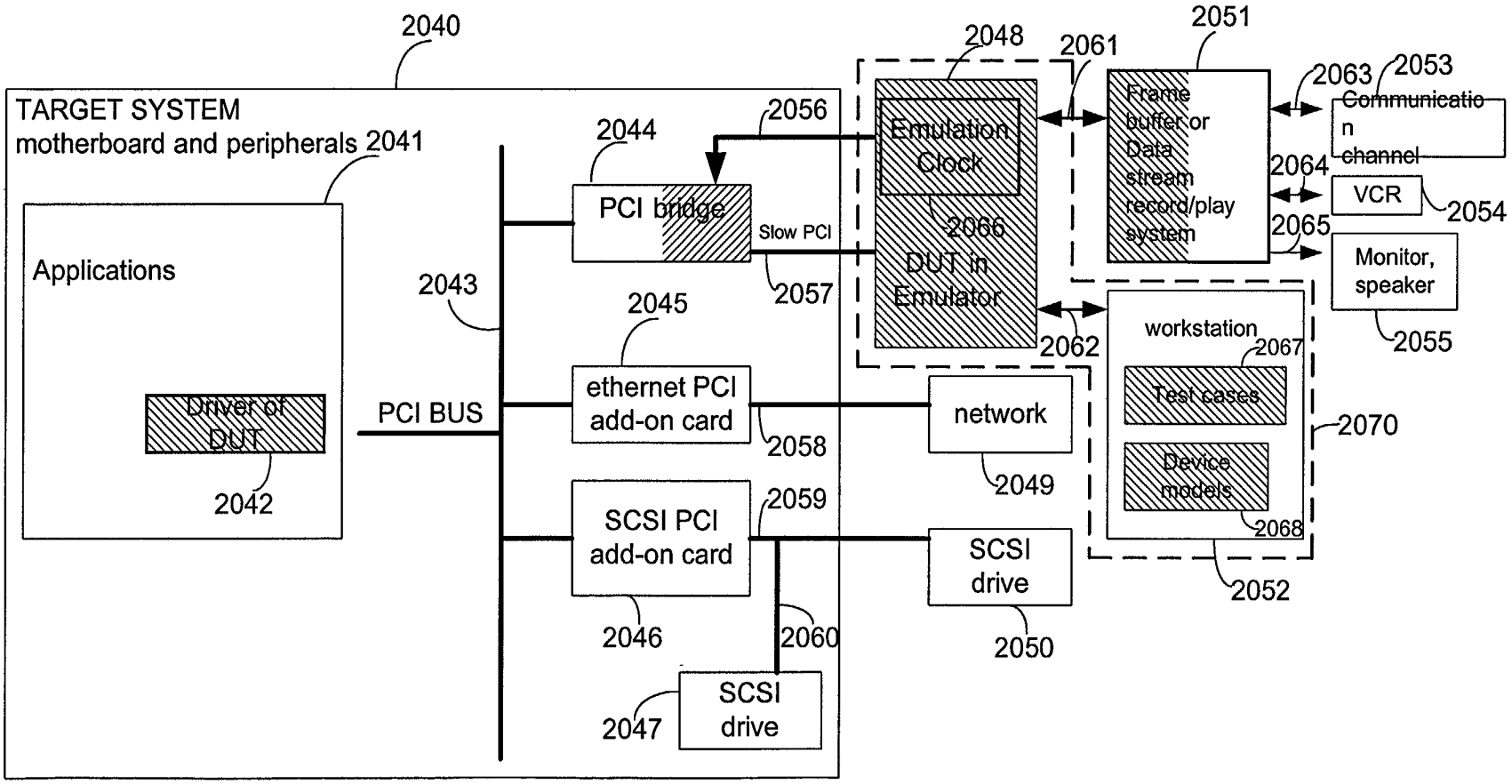
Typical Hardware/Software Co-Verification



: DUT (Device Under Test)

FIG. 65

Typical Co-Verification by Using Emulator



: **running time at emulation speed**

The rest of the target system is running at full speed.

FIG. 66

SIMULATION

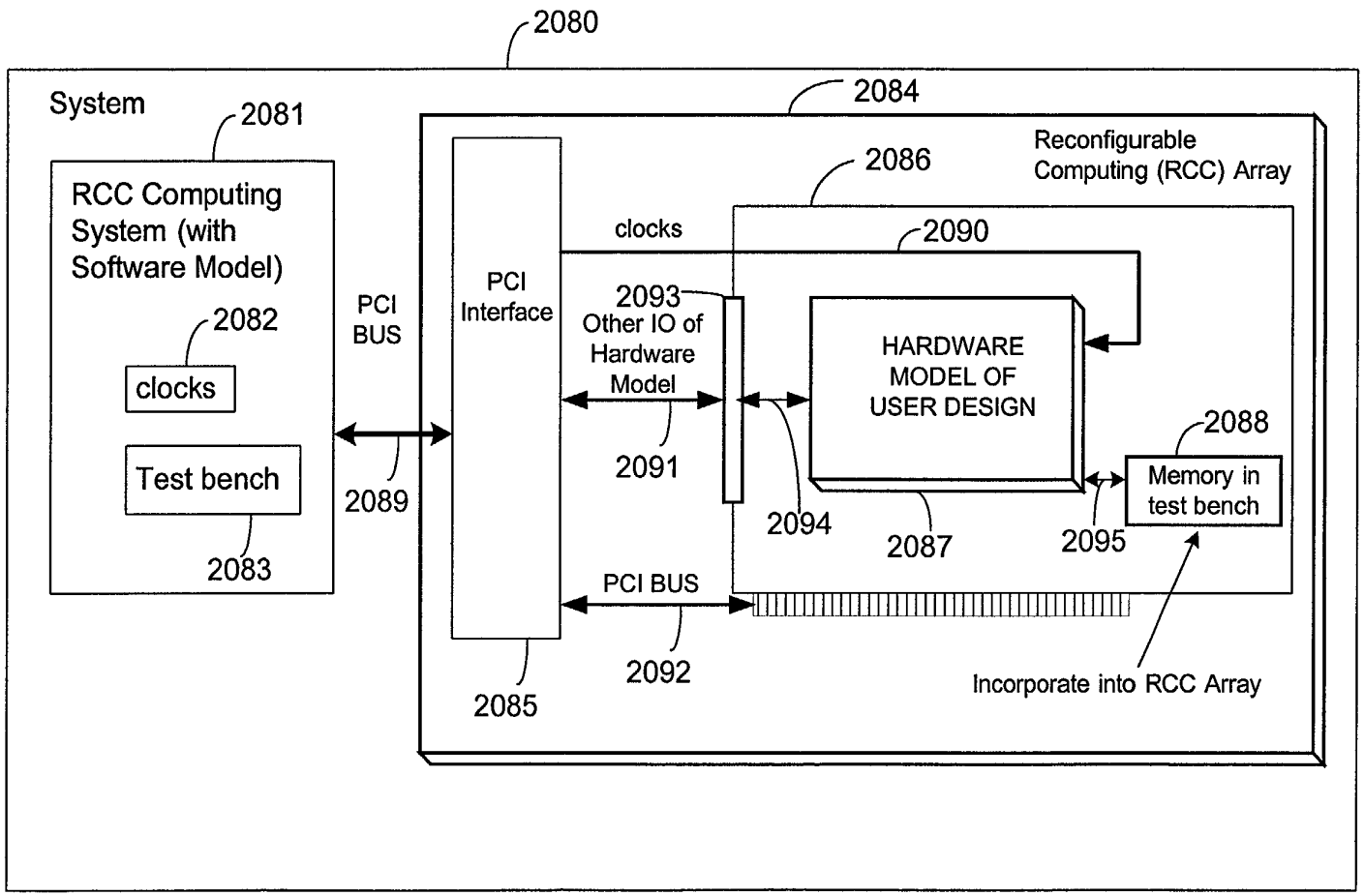


FIG. 67

CO-VERIFICATION WITHOUT EXTERNAL I/O

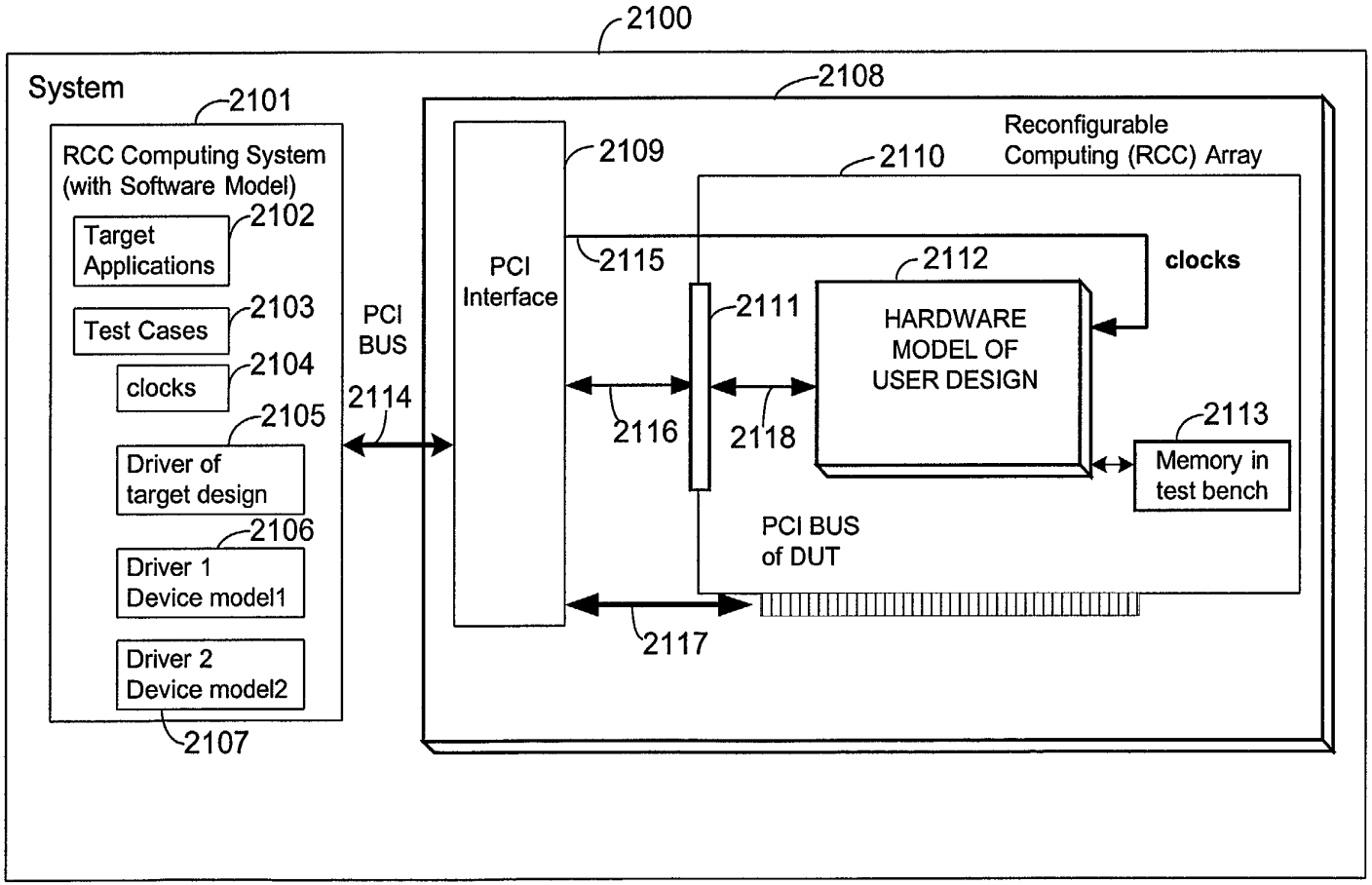


FIG. 68

CO-VERIFICATION WITH EXTERNAL I/O

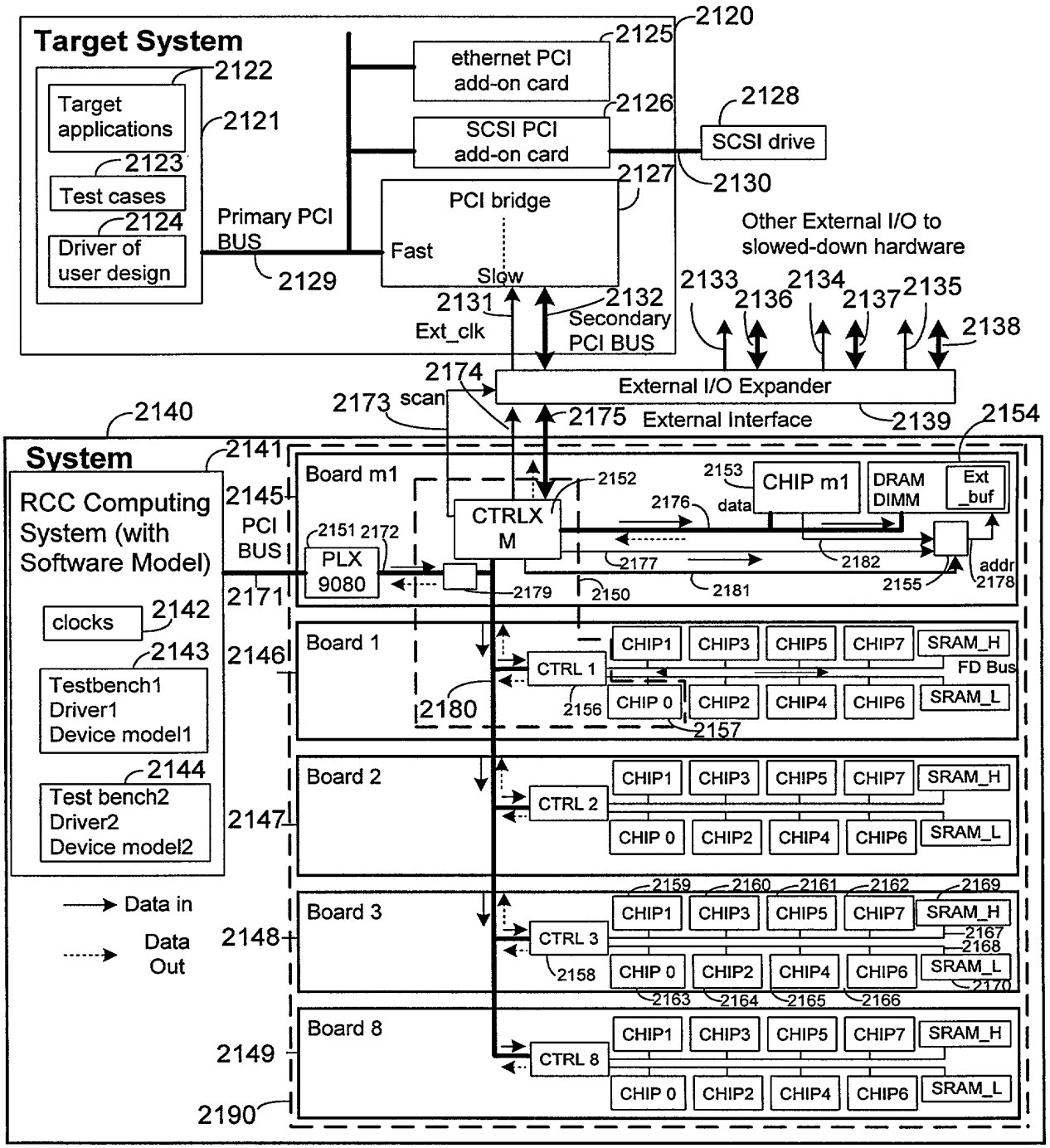


FIG. 69

CONTROL OF DATA-IN CYCLE

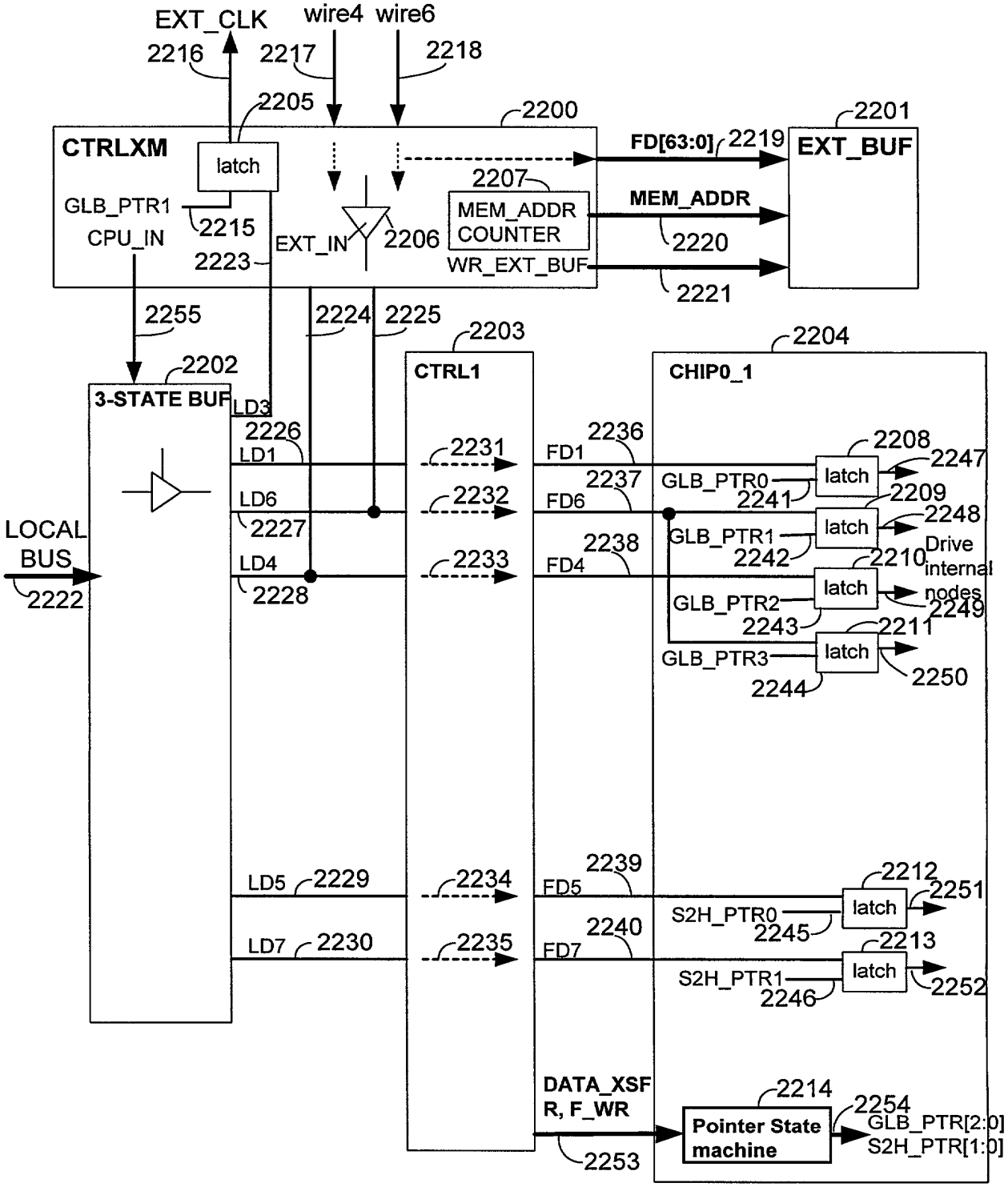


FIG. 70

CONTROL OF DATA-OUT CYCLE

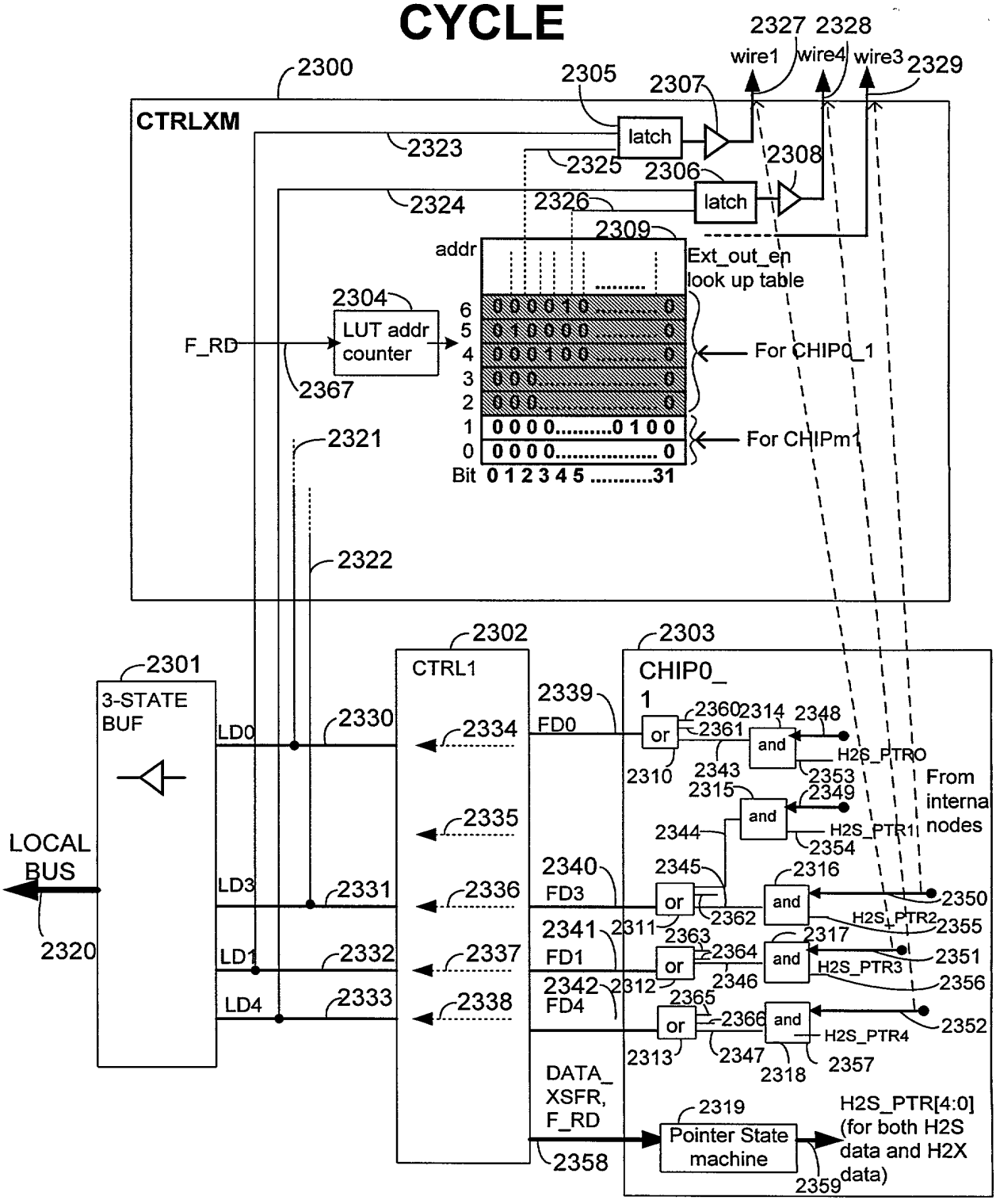


FIG. 71

CONTROL OF DATA-IN CYCLE

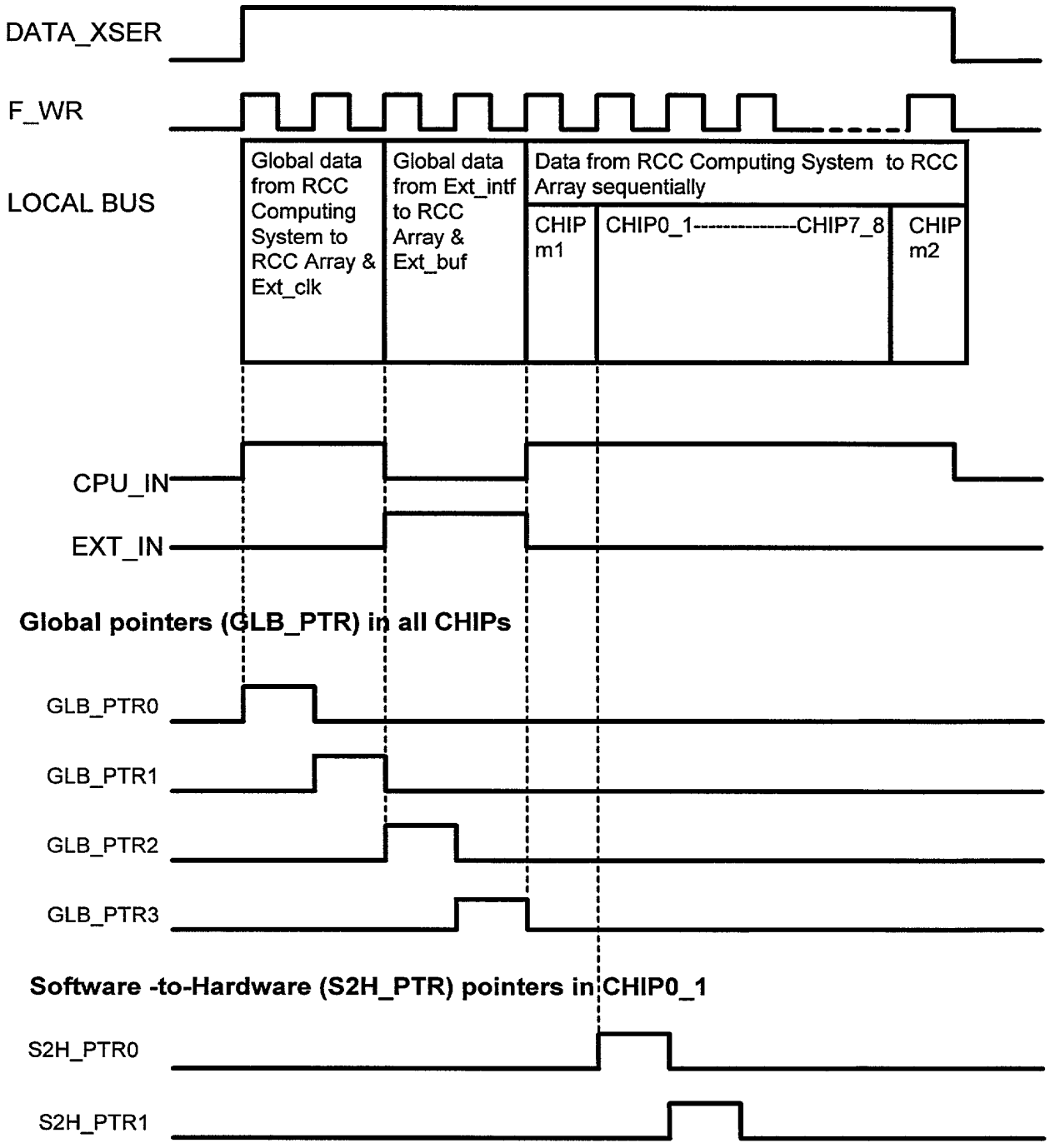


FIG. 72

CONTROL OF DATA-OUT CYCLE

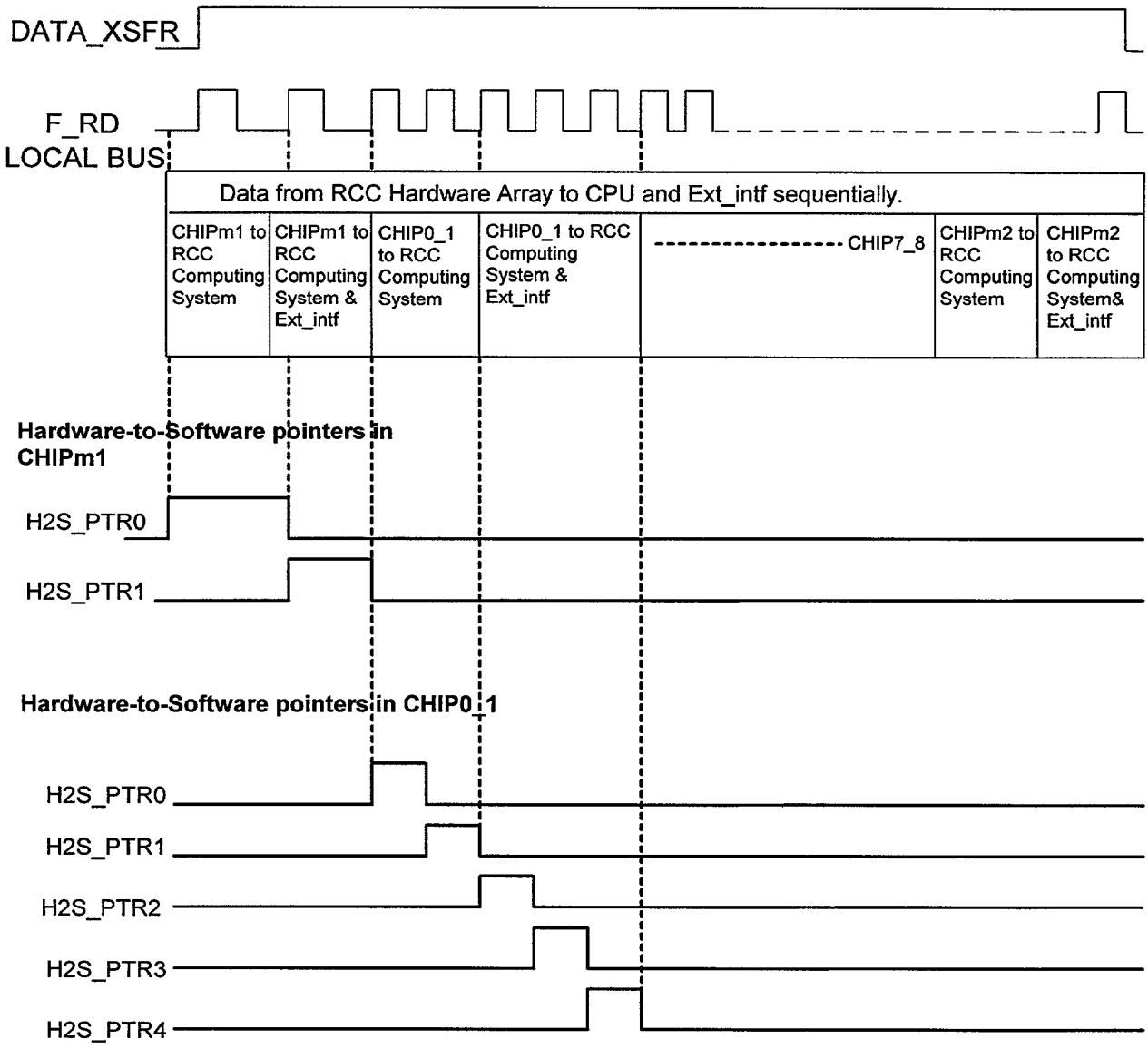


FIG. 73

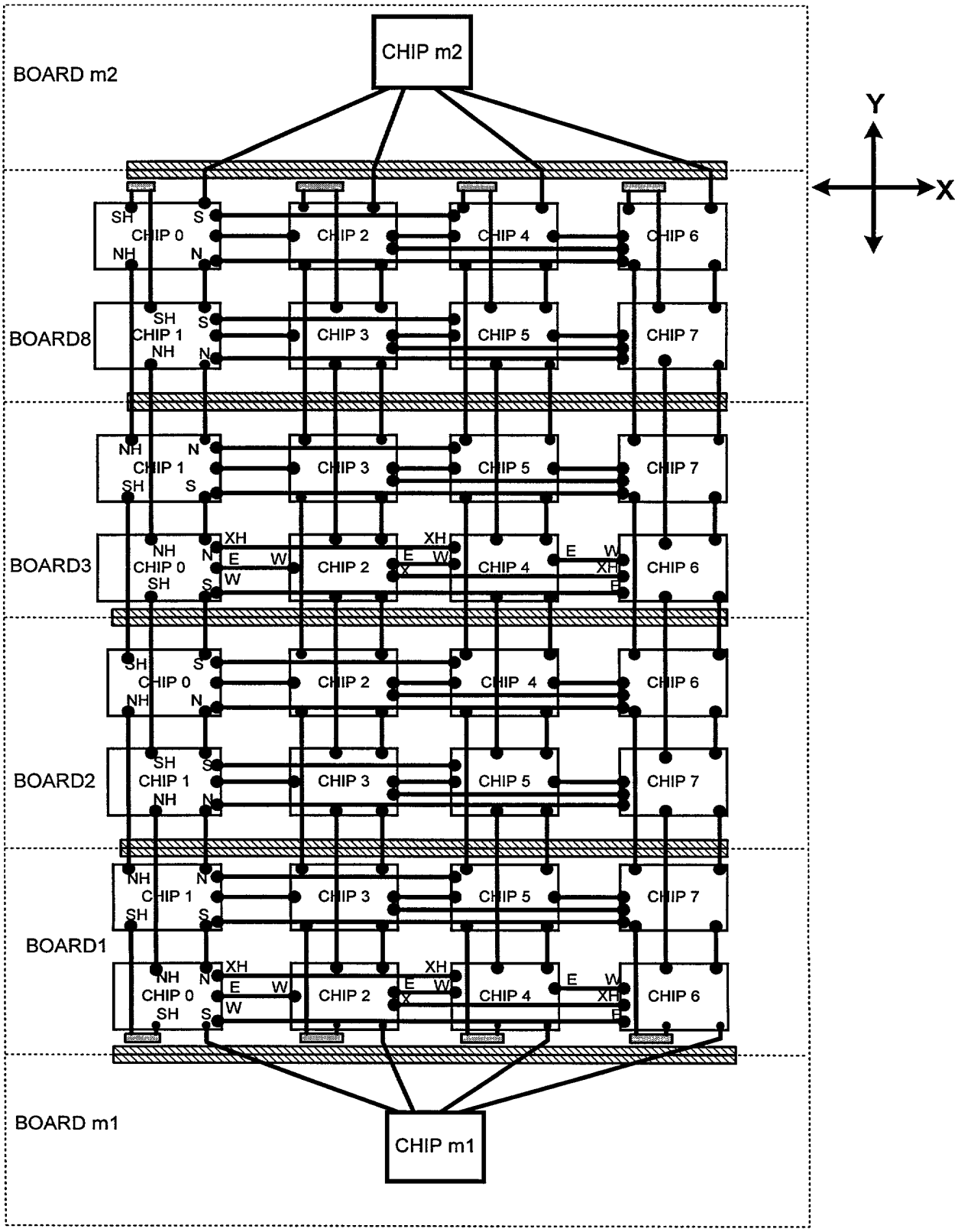

CHIP m2
BOARD m2
Y
X
SH
S
CHIP 0
NH
N
CHIP 2
CHIP 4
CHIP 6
BOARD8
CHIP 1
CHIP 3
CHIP 5
CHIP 7
BOARD3
XH
E
W
BOARD2
BOARD1
BOARD m1
CHIP m1

FIG. 74

SHIFT REGISTER

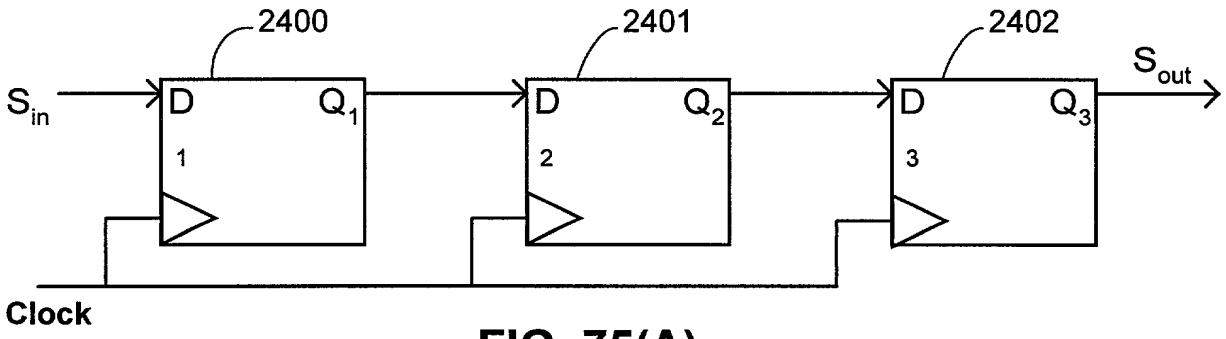


FIG. 75(A)

HOLD TIME ASSUMPTION FOR SHIFT REGISTER

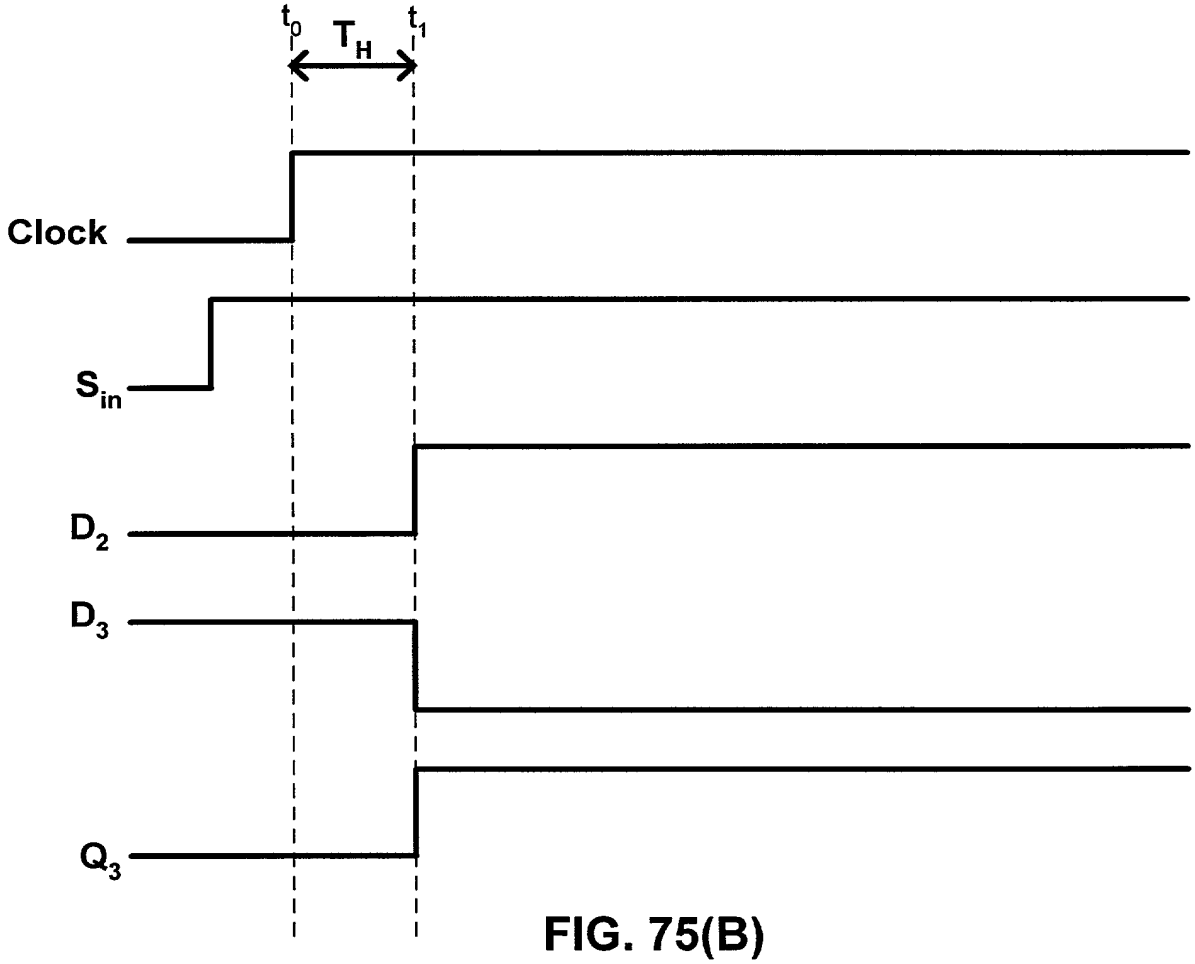


FIG. 75(B)

MULTIPLE FPGA MAPPING FOR SHIFT REGISTER

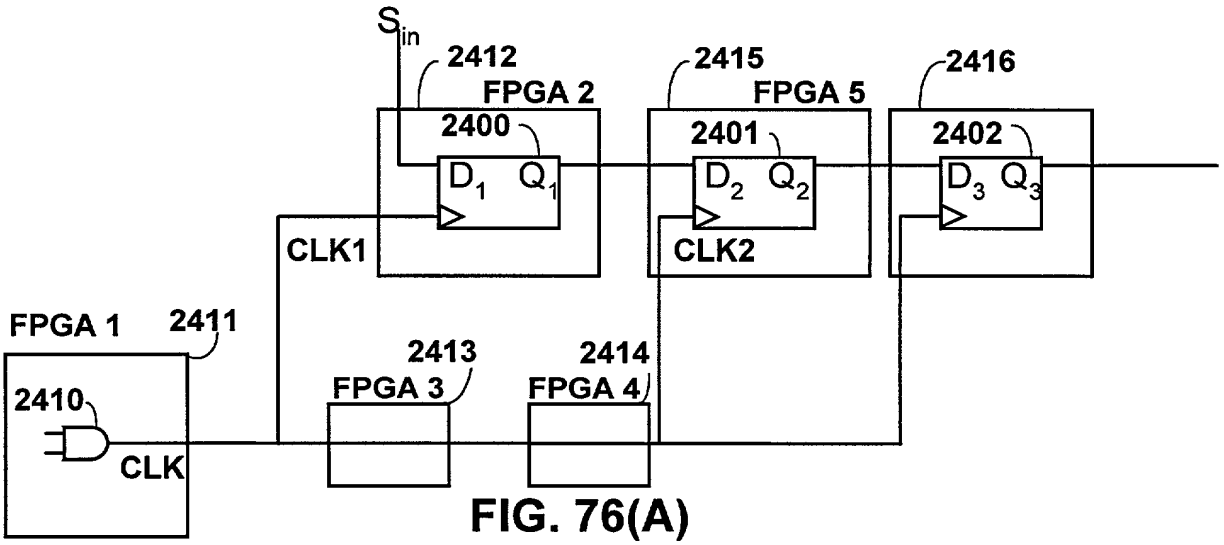


FIG. 76(A)

HOLD TIME VIOLATION BY LONG CLOCK SKEW

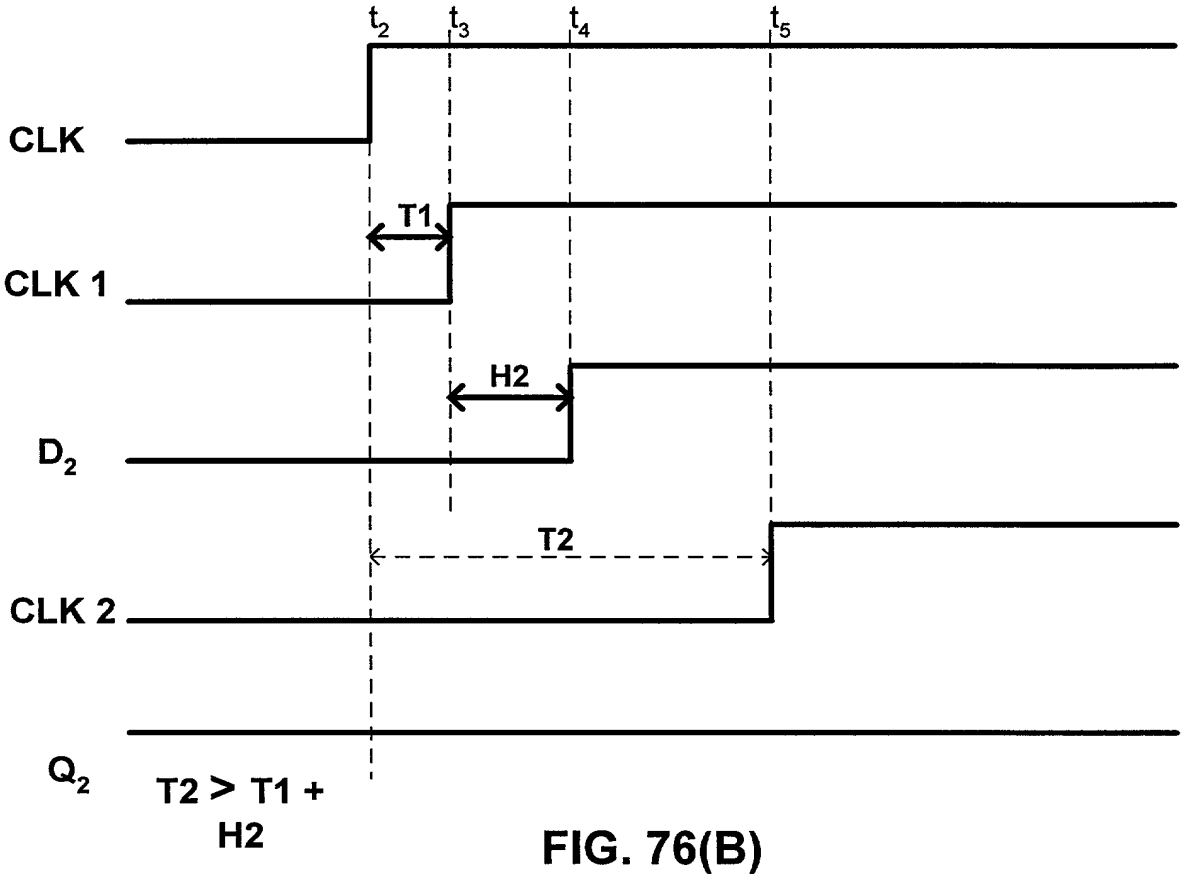


FIG. 76(B)

CLOCK GLITCH PROBLEM

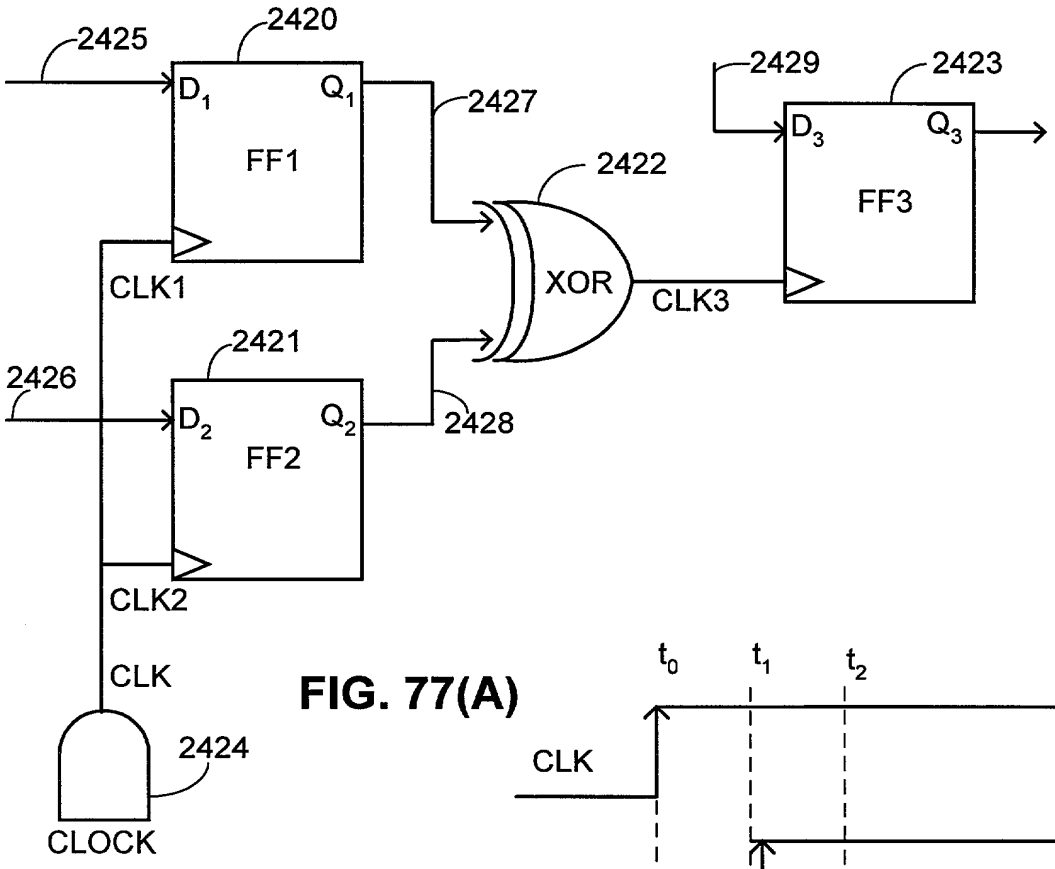


FIG. 77(A)

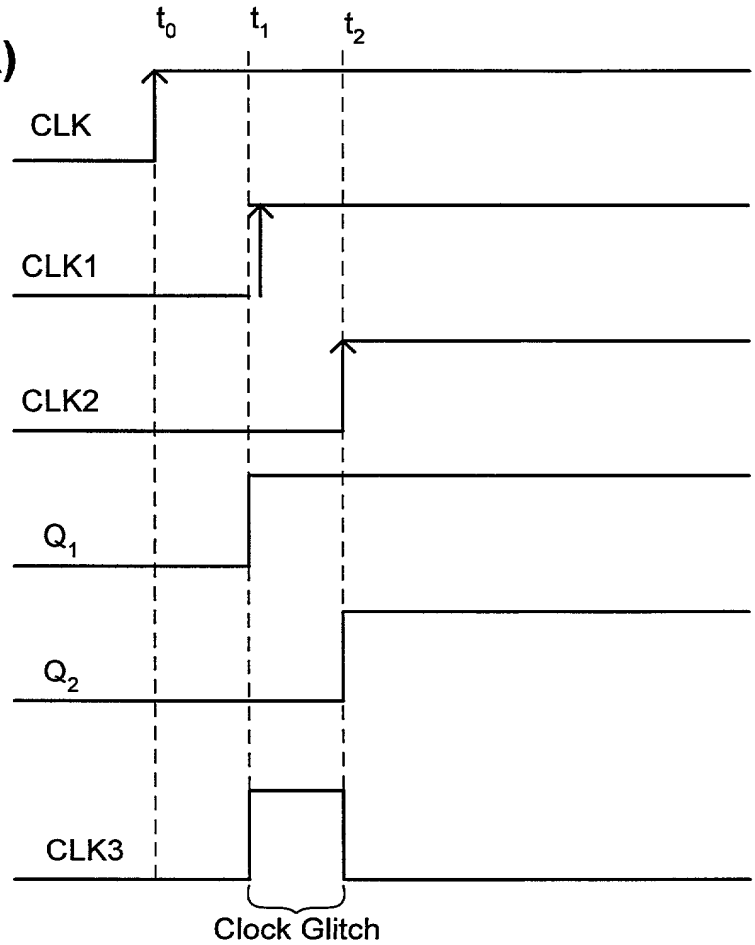
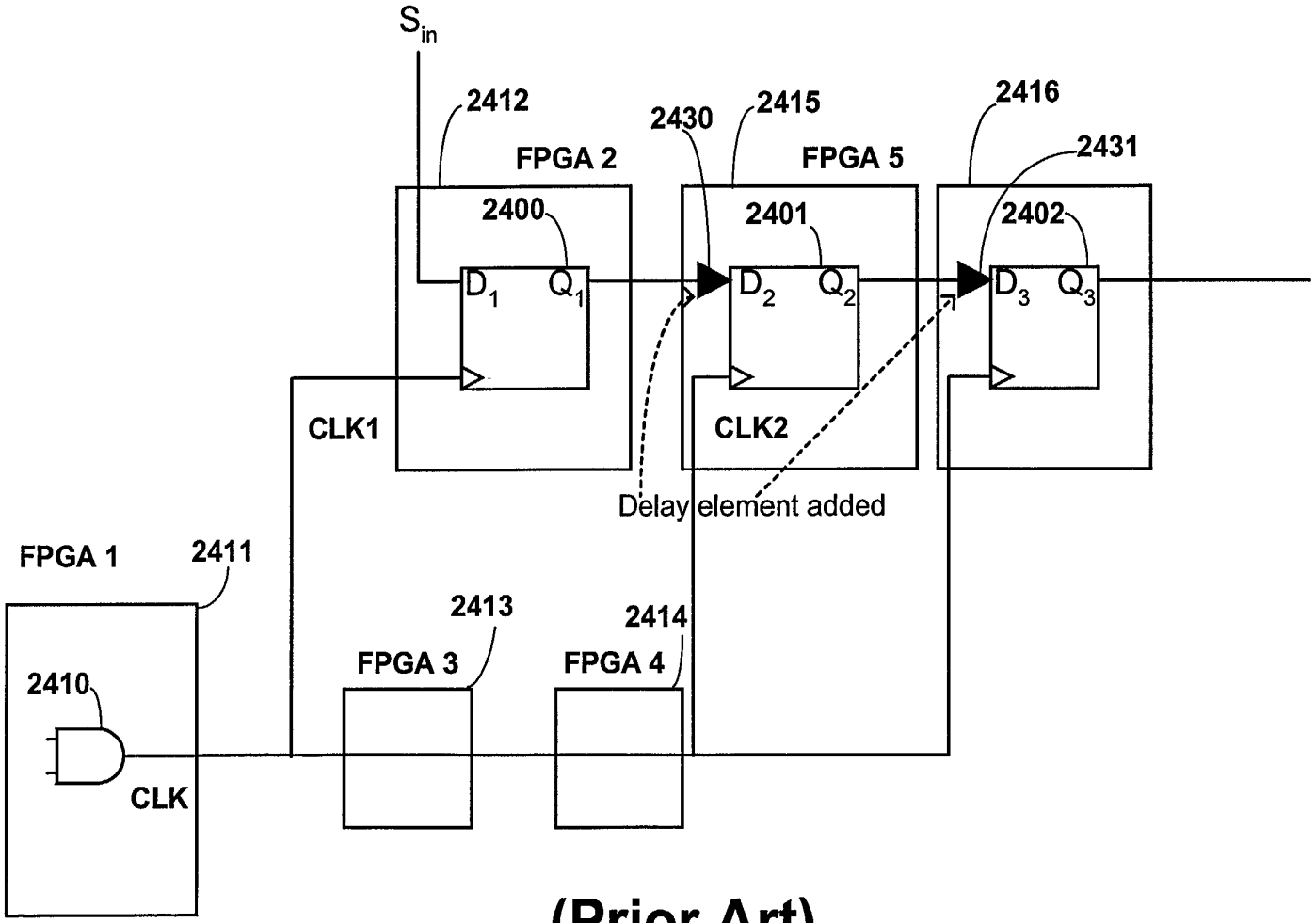


FIG. 77(B)

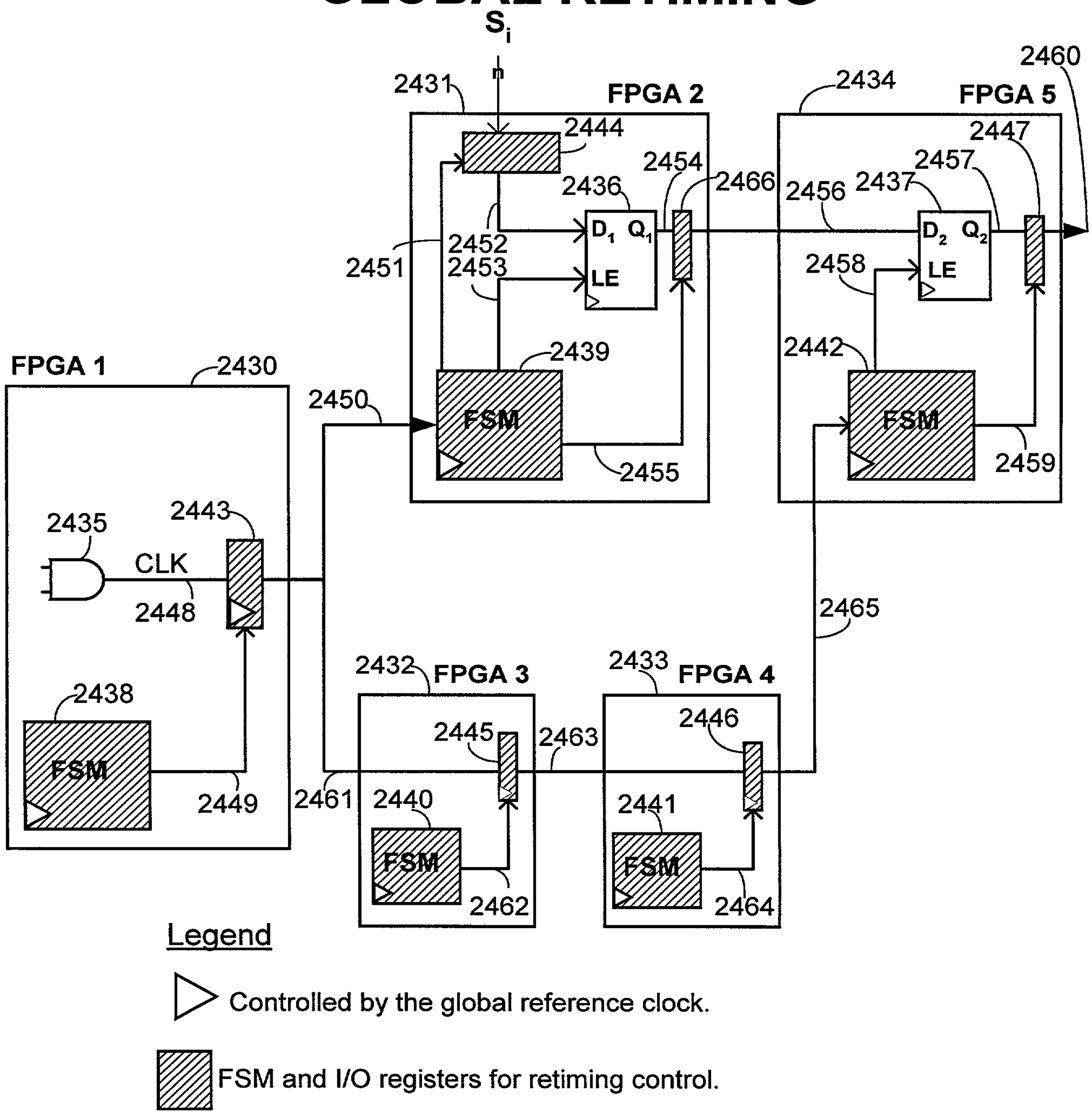
TIMING ADJUSTMENT BY ADDING DELAY



(Prior Art)

FIG. 78

GLOBAL RETIMING



(Prior Art)

FIG. 79

TIGF LATCH

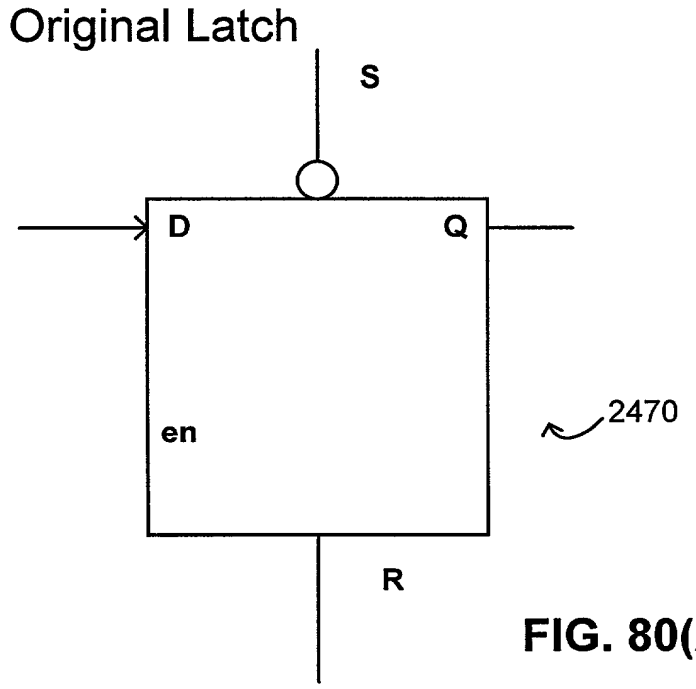


FIG. 80(A)

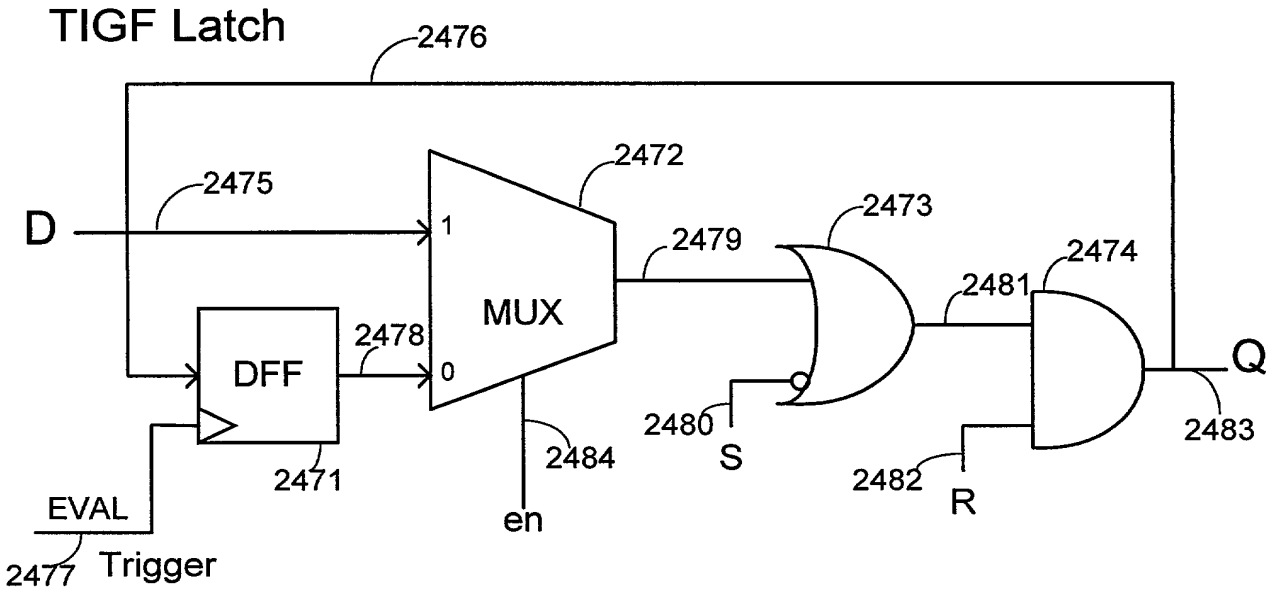


FIG. 80(B)

TIGF DFF

Original DFF

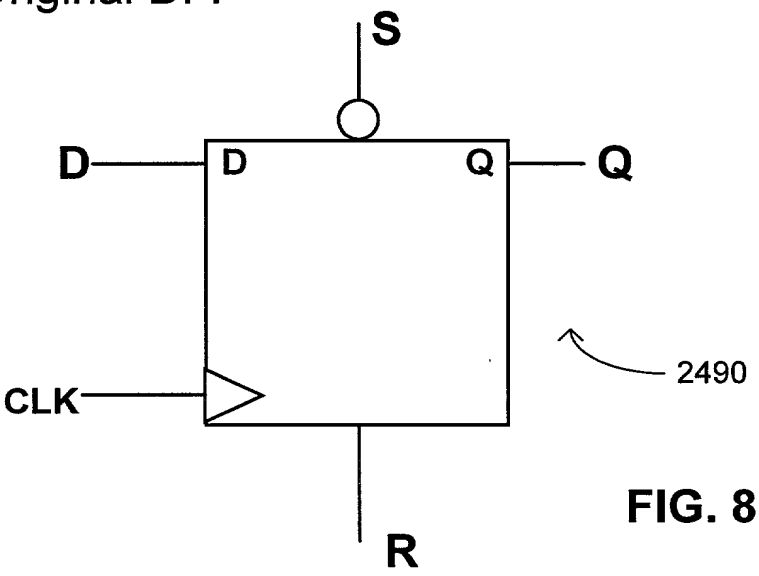


FIG. 81(A)

TIGF DFF and Edge Detector

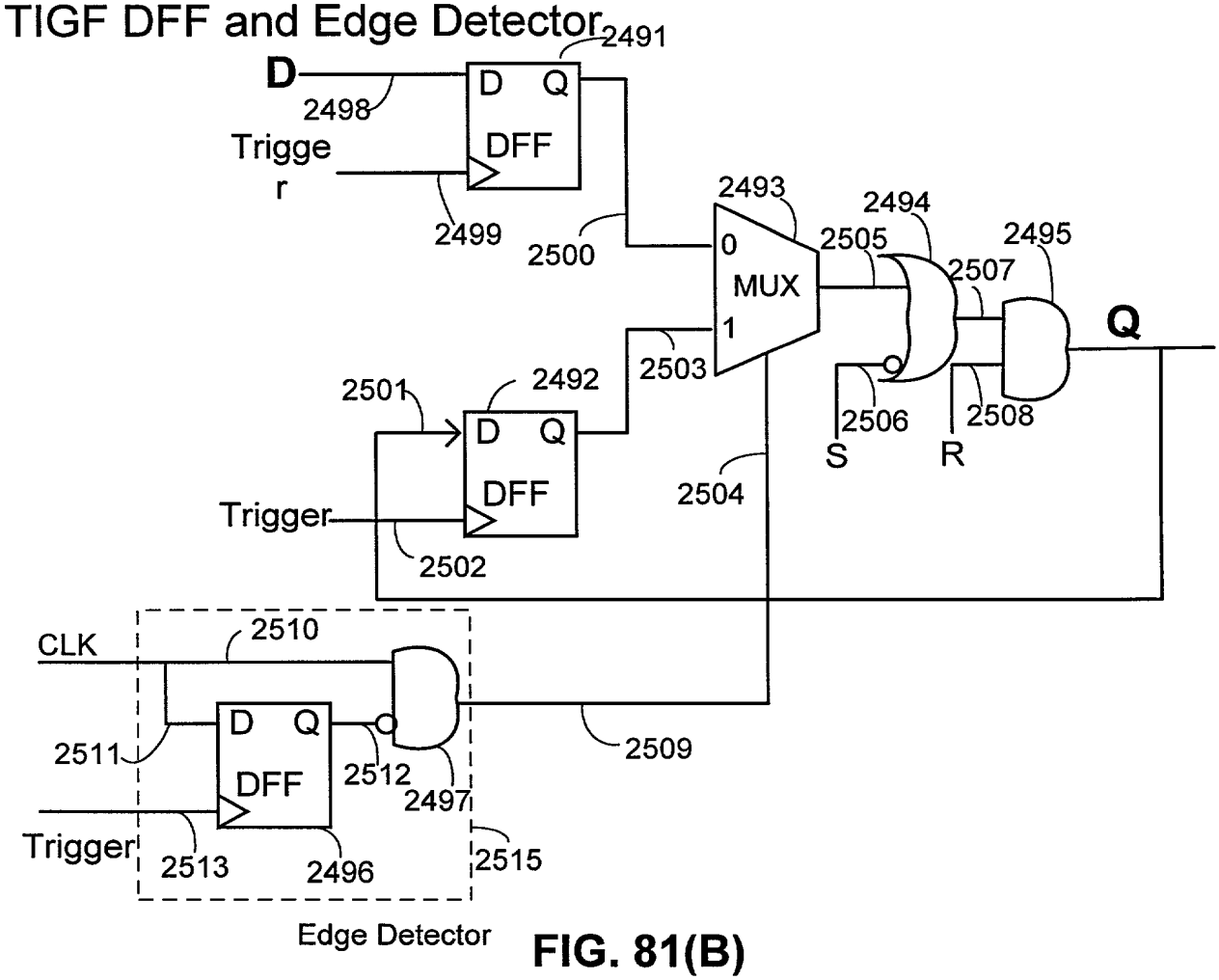


FIG. 81(B)

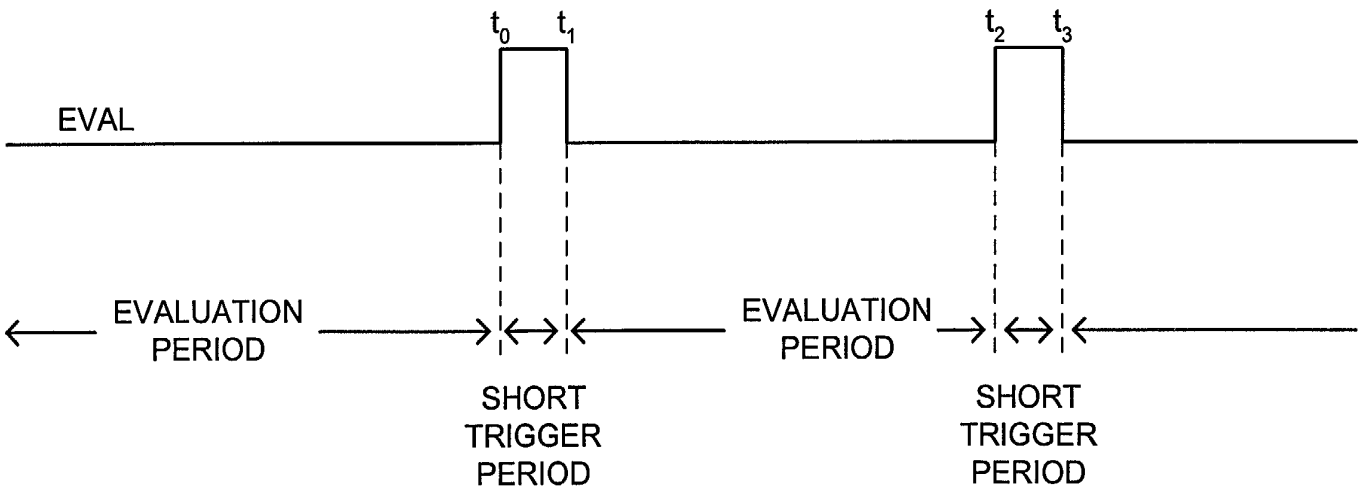

GLOBAL TRIGGER SIGNAL
t_0
t_1
t_2
t_3
EVAL
EVALUATION PERIOD
EVALUATION PERIOD
SHORT TRIGGER PERIOD
SHORT TRIGGER PERIOD

FIG. 82

RCC System

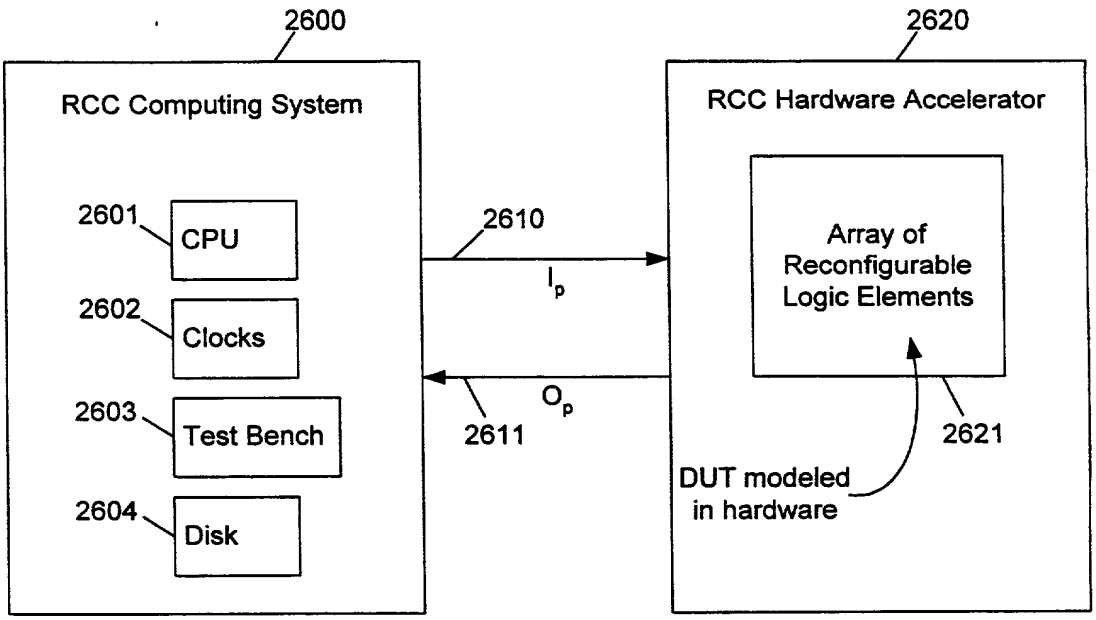


FIG. 83

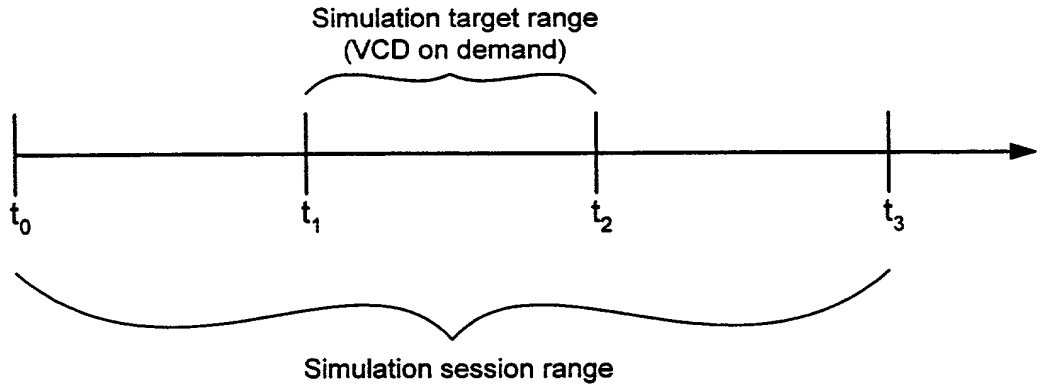

Simulation target range
(VCD on demand)
t_0
t_1
t_2
t_3
Simulation session range

FIG. 84

SINGLE-ROW FPGA PER BOARD

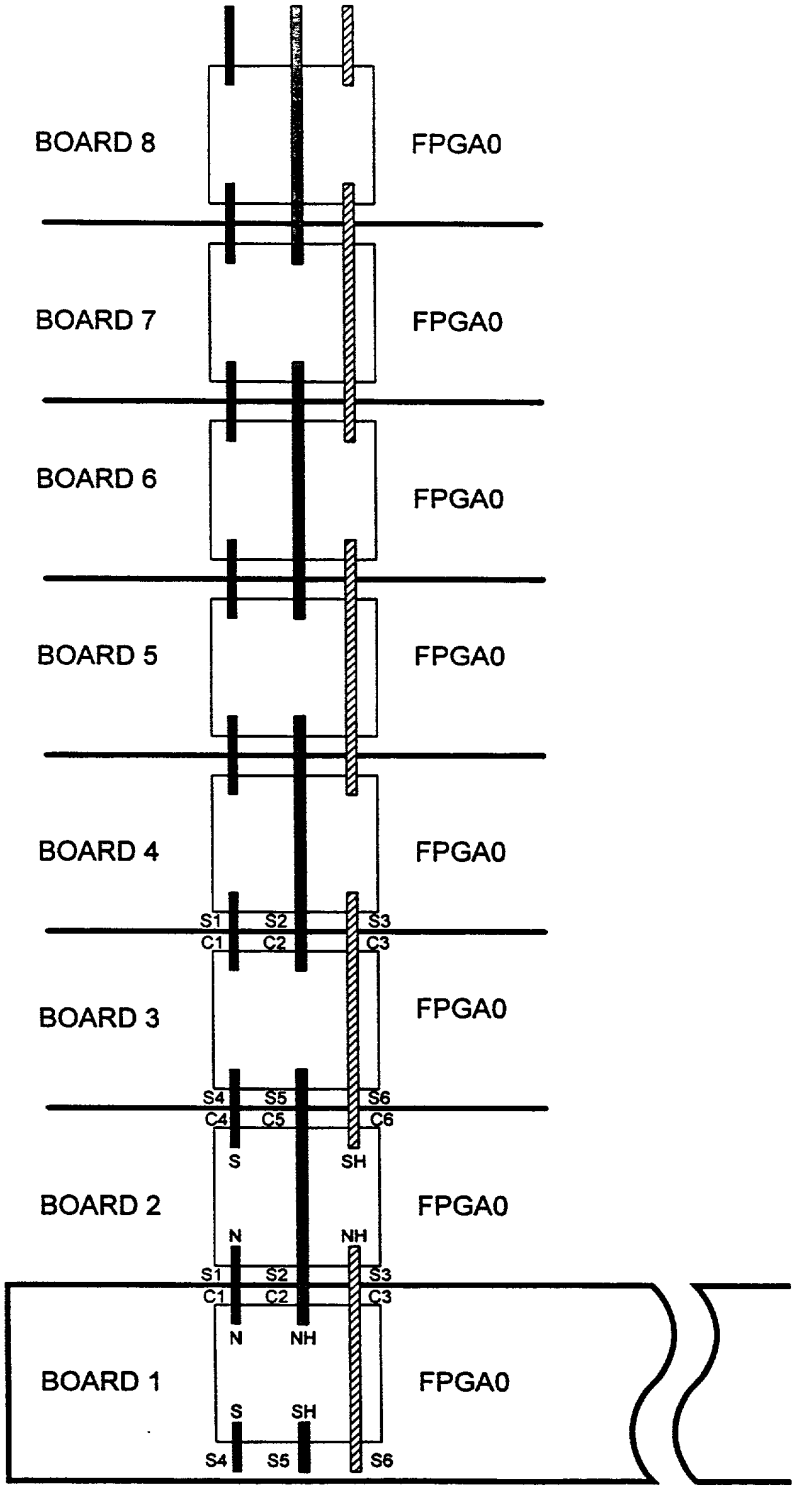


FIG. 85

TWO-ROW FPGA PER BOARD

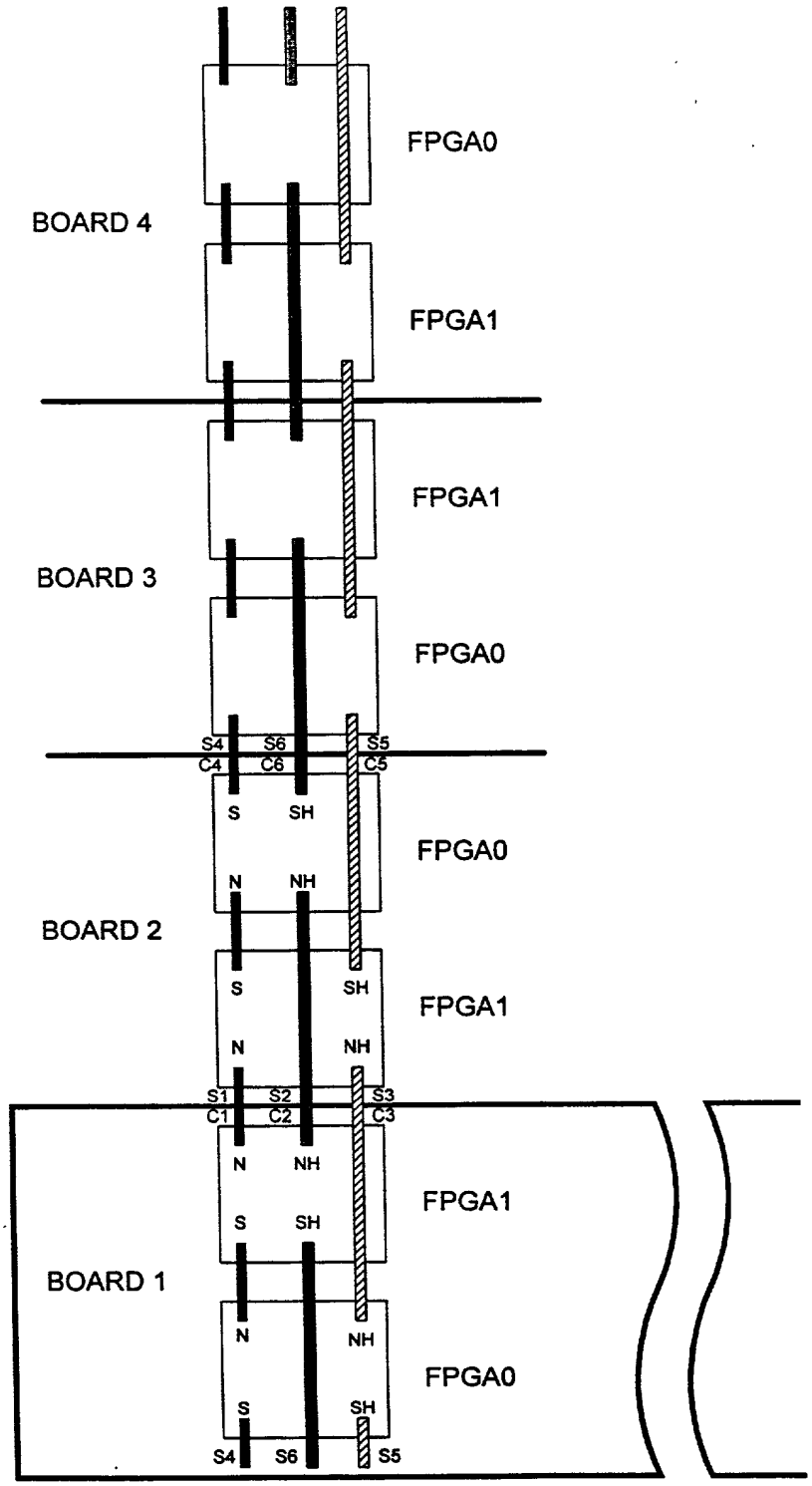


FIG. 86

THREE-ROW FPGA PER BOARD

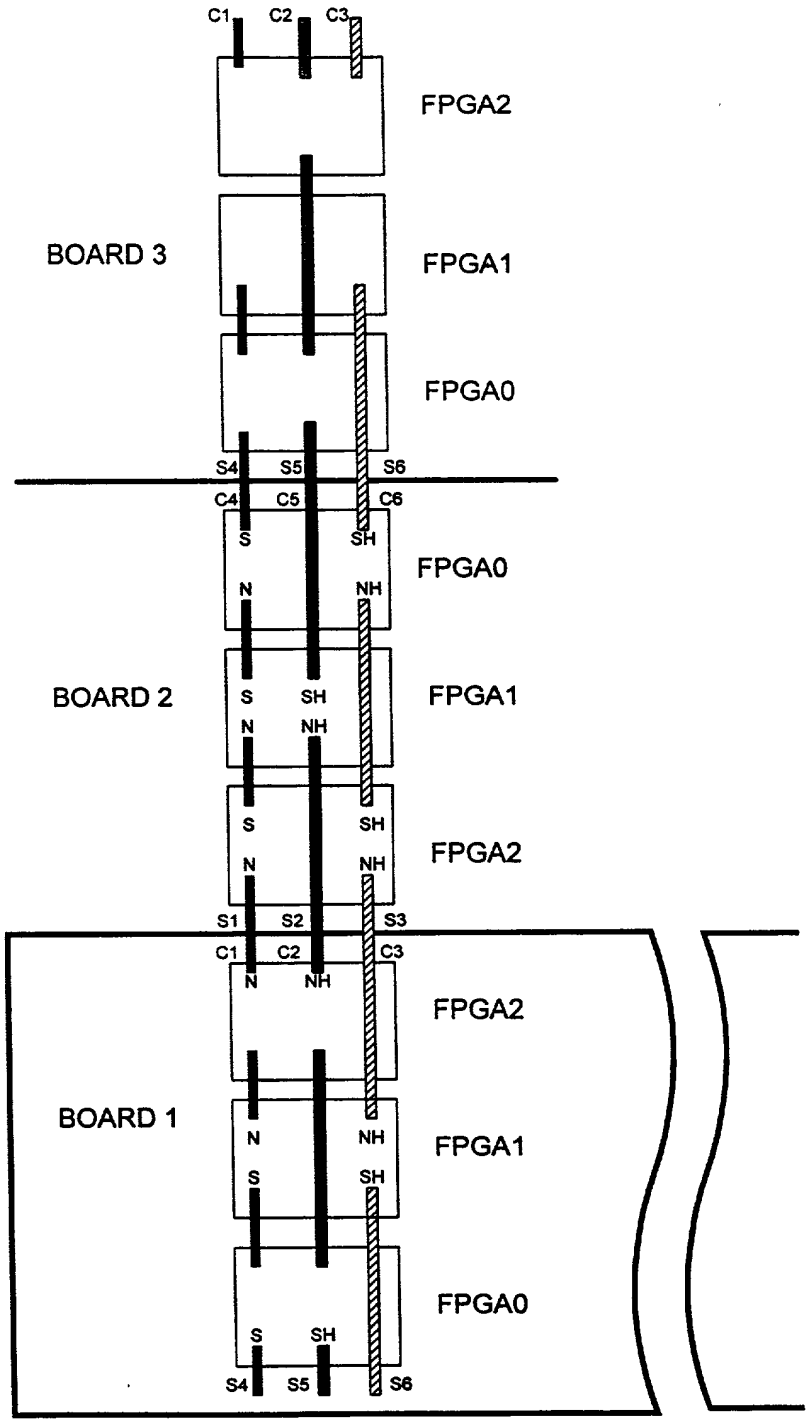


FIG. 87

FOUR-ROW FPGA PER BOARD

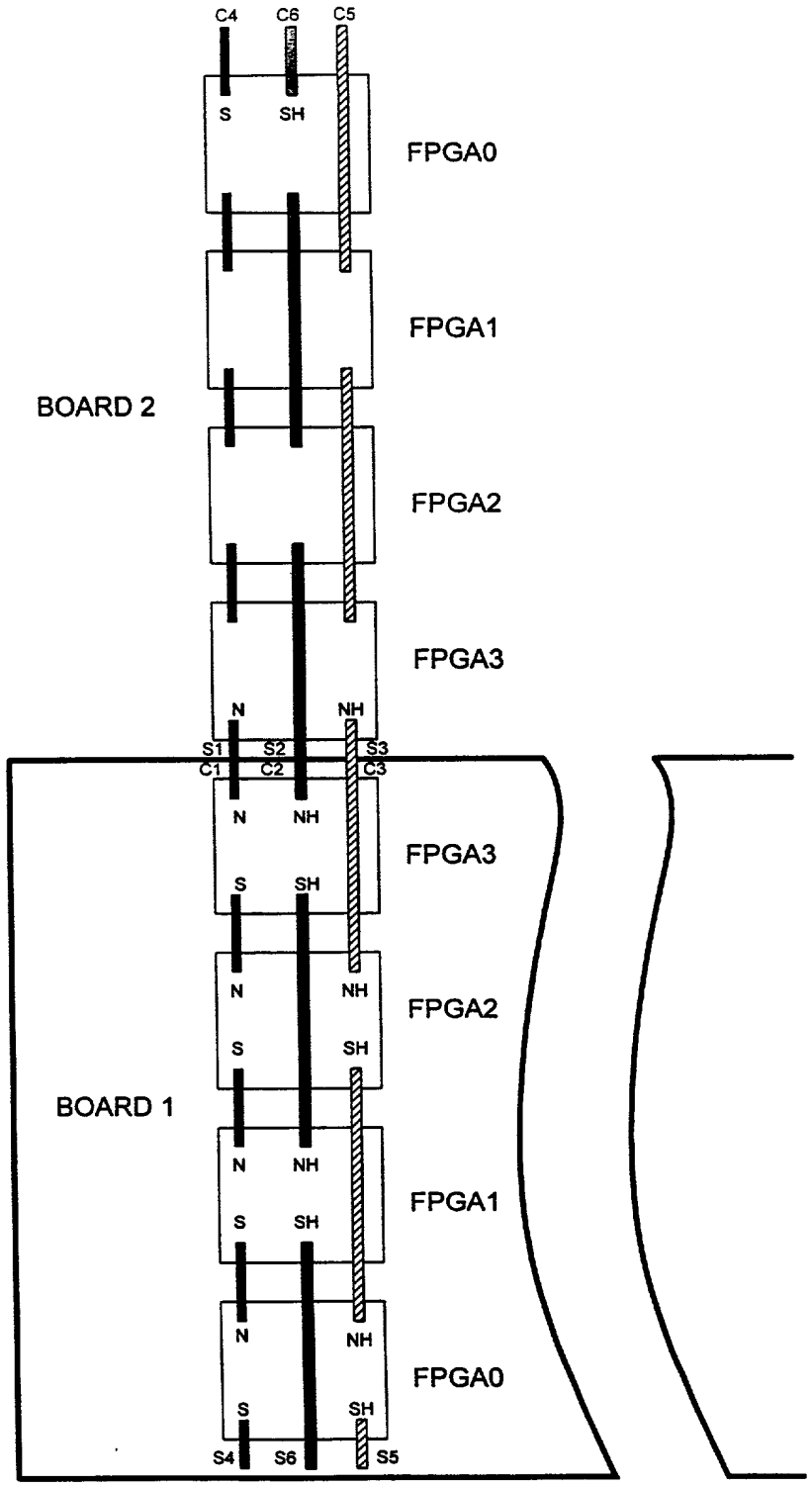


FIG. 88

INTERCONNECT FOR THREE-ROW PER BOARD

I/O Signals	Odd Board	Even Board	Common Board
	Connector-Group Pin-position	Connector-Group Pin-position	Connector-Group Pin-position
FPGA2_N	C1	S1	C1, S1
FPGA2_NH	C2	S3	C2, S3
FPGA1_NH	C3	S2	C3, S2
FPGA0_S	S4	C4	C4, S4
FPGA0_SH	S5	C6	C6, S5
FPGA1_SH	S6	C5	C5, S6

FIG. 89

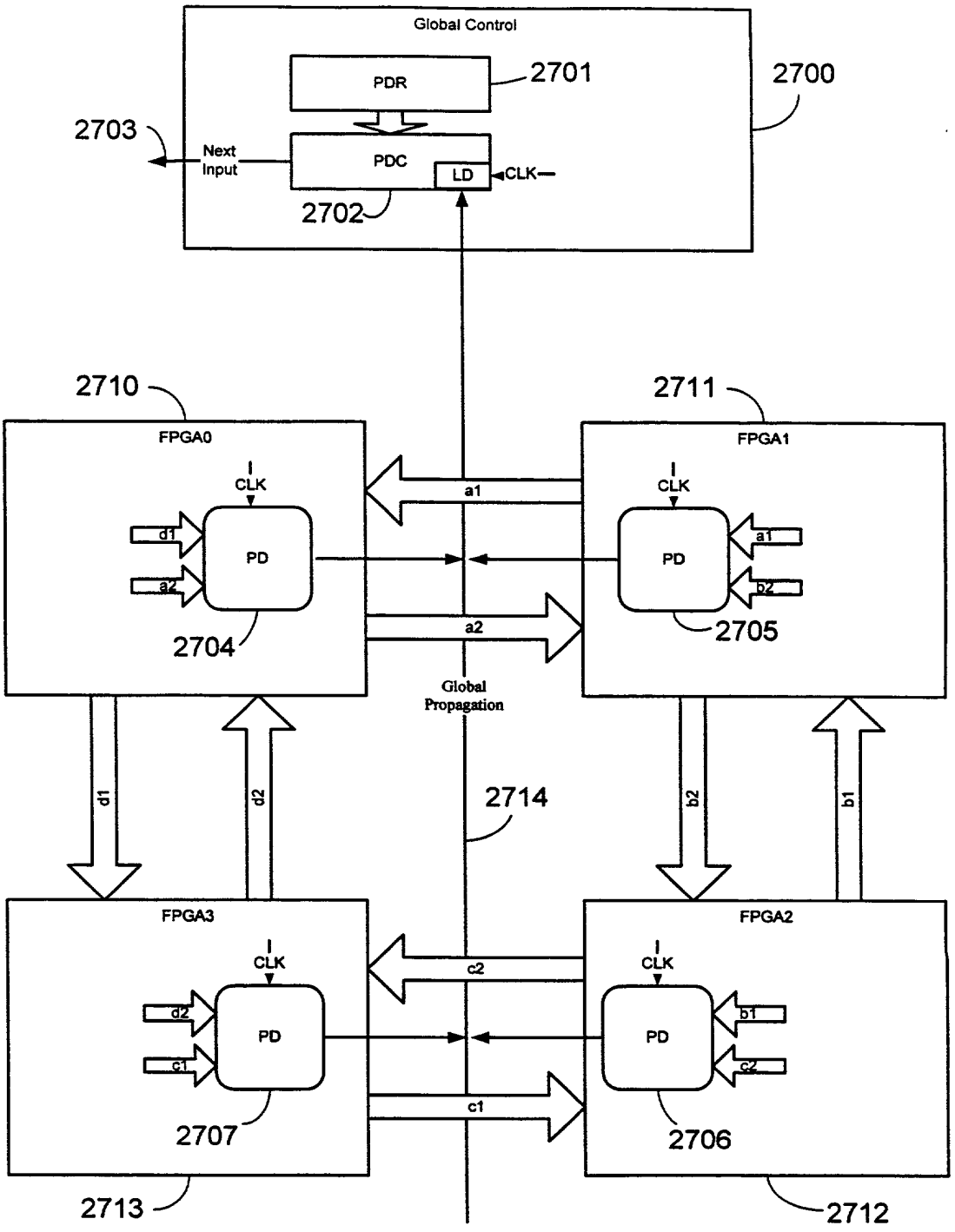

Global Control
PDR
2701
2700
2703
Next Input
PDC
LD
CLK
2702
2710
2711
FPGA0
FPGA1
CLK
d1
a2
PD
a1
b2
2704
2705
Global Propagation
2714
d2
b1
FPGA3
FPGA2
c2
c1
2707
2706
2713
2712

FIG. 90

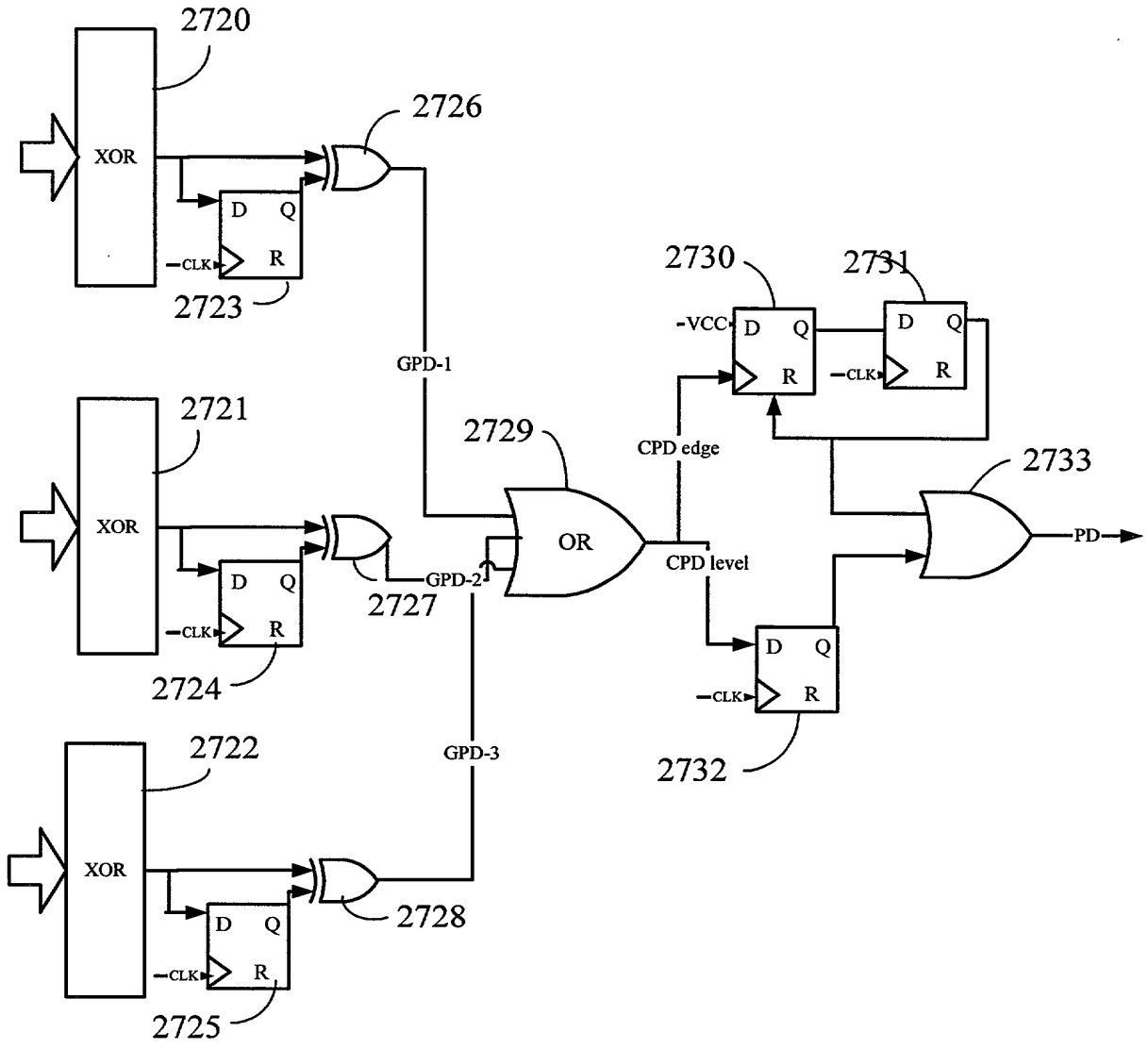

2720
XOR
2726
D Q
CLK R
2723
GPD-1
2721
XOR
2729
OR
2730
-VCC
D Q
R
2731
D Q
CLK R
CPD edge
2733
PD
D Q
CLK R
2727
GPD-2
CPD level
2724
2722
XOR
GPD-3
D Q
CLK R
2728
2732
2725

FIG. 91

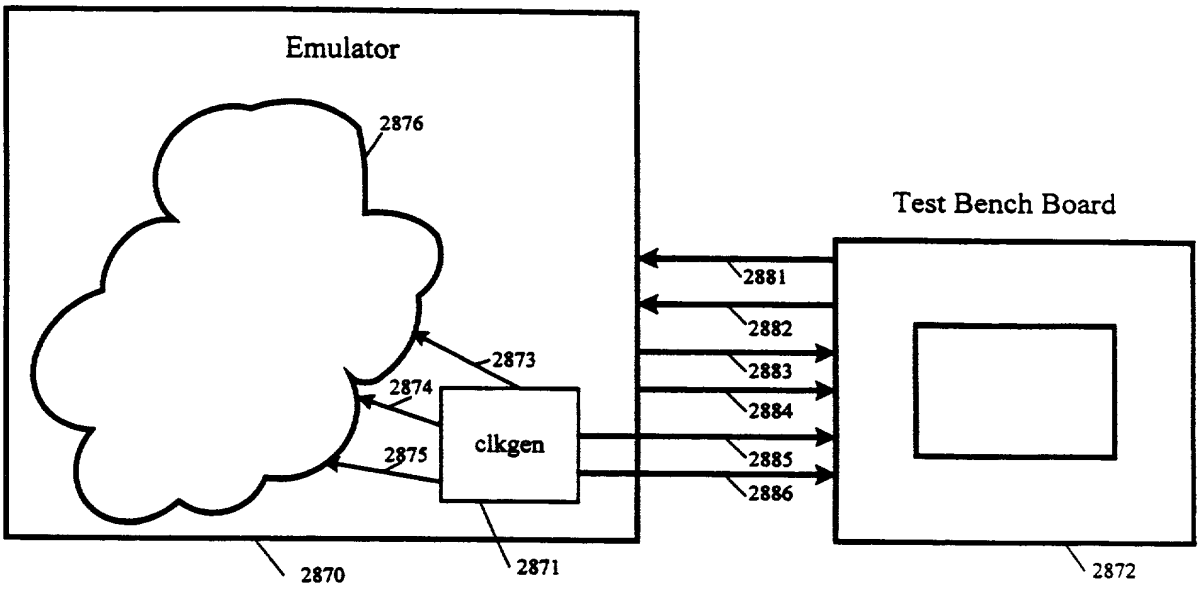

Emulator
2876
Test Bench Board
2881
2882
2883
2884
2885
2886
2873
2874
clkgen
2875
2870
2871
2872

FIG. 92

Clock Specification

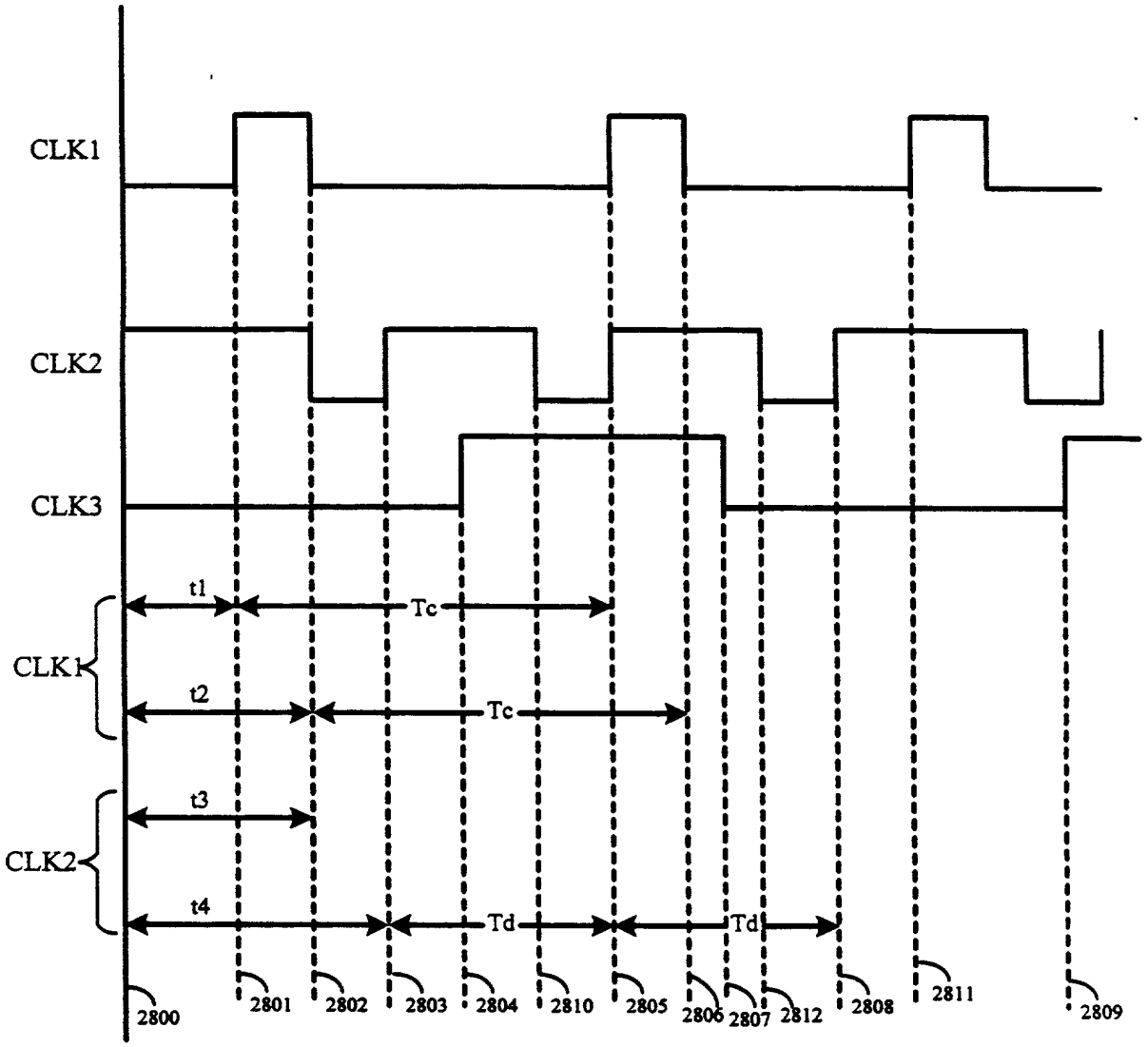


FIG. 93

Clock Generation Scheduler w/ Slices

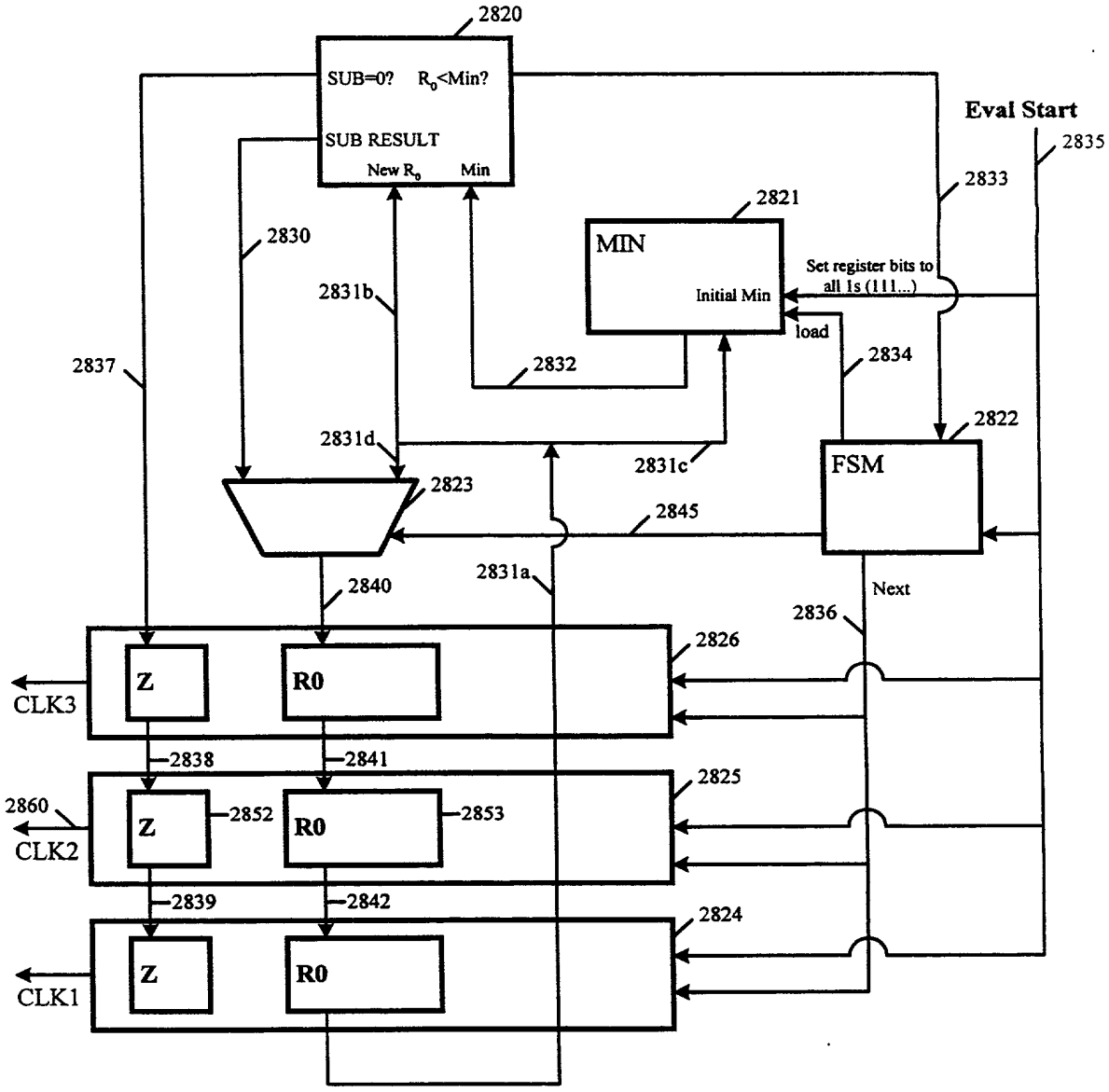


FIG. 94

Clock Generation Slice

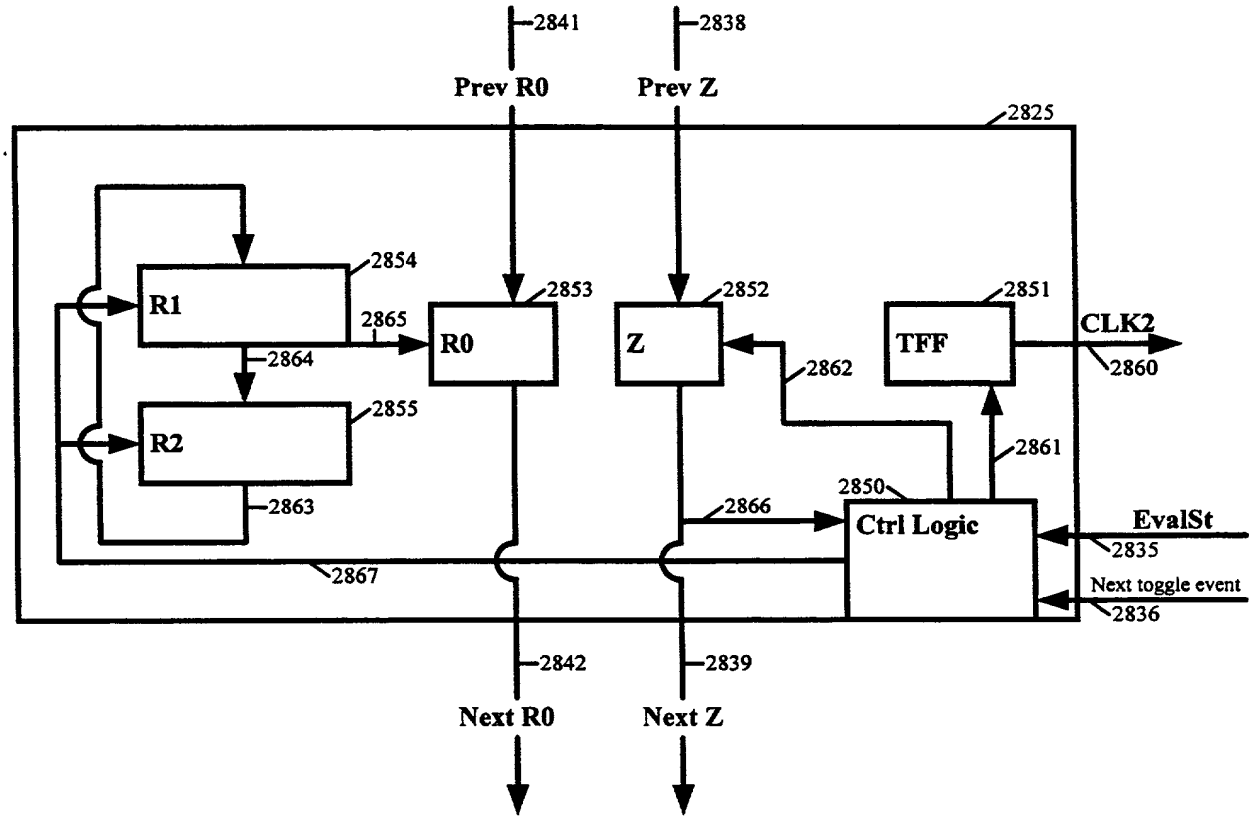


FIG. 95

Clock Generation Scheduler and Slices

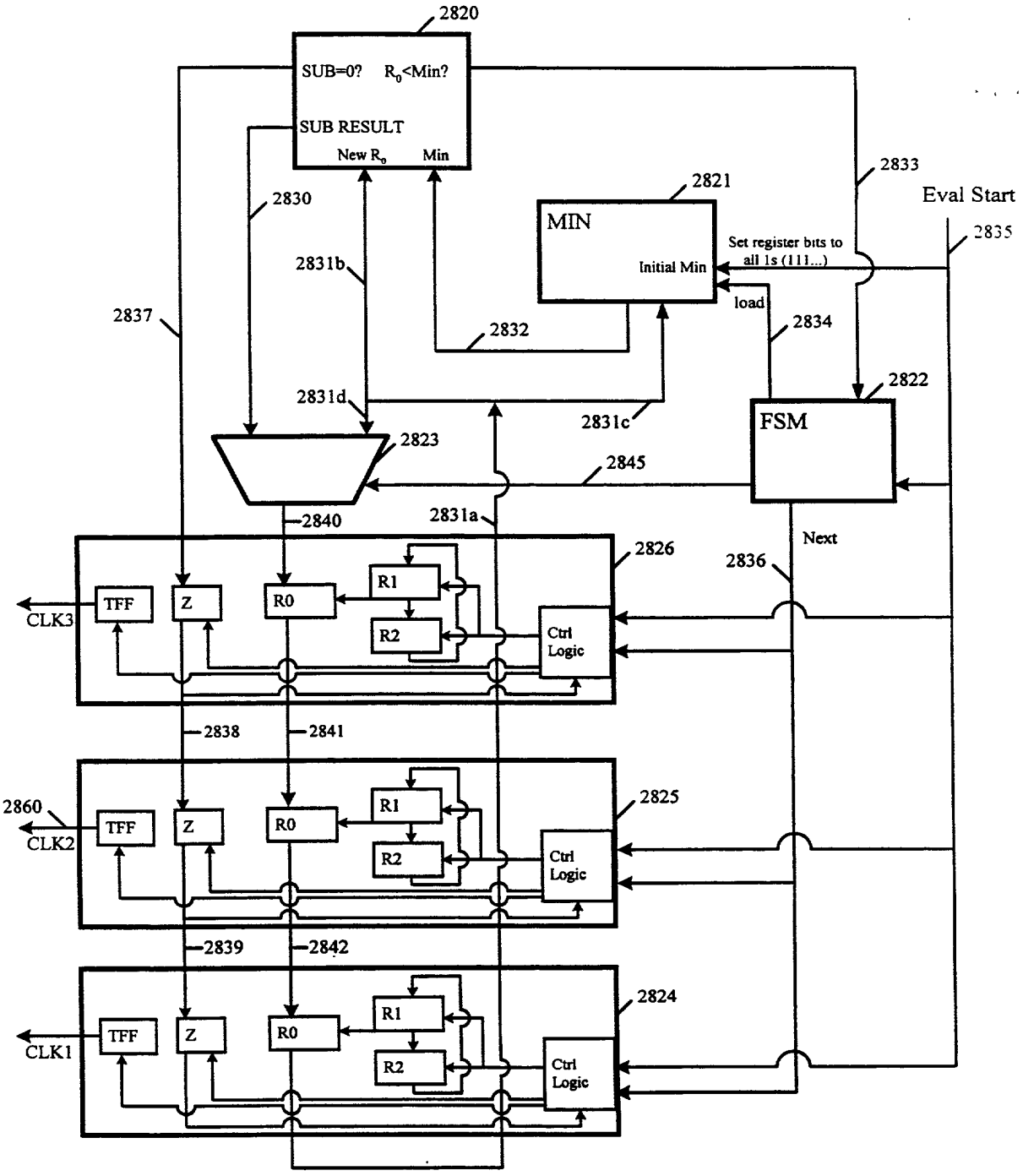


FIG. 96

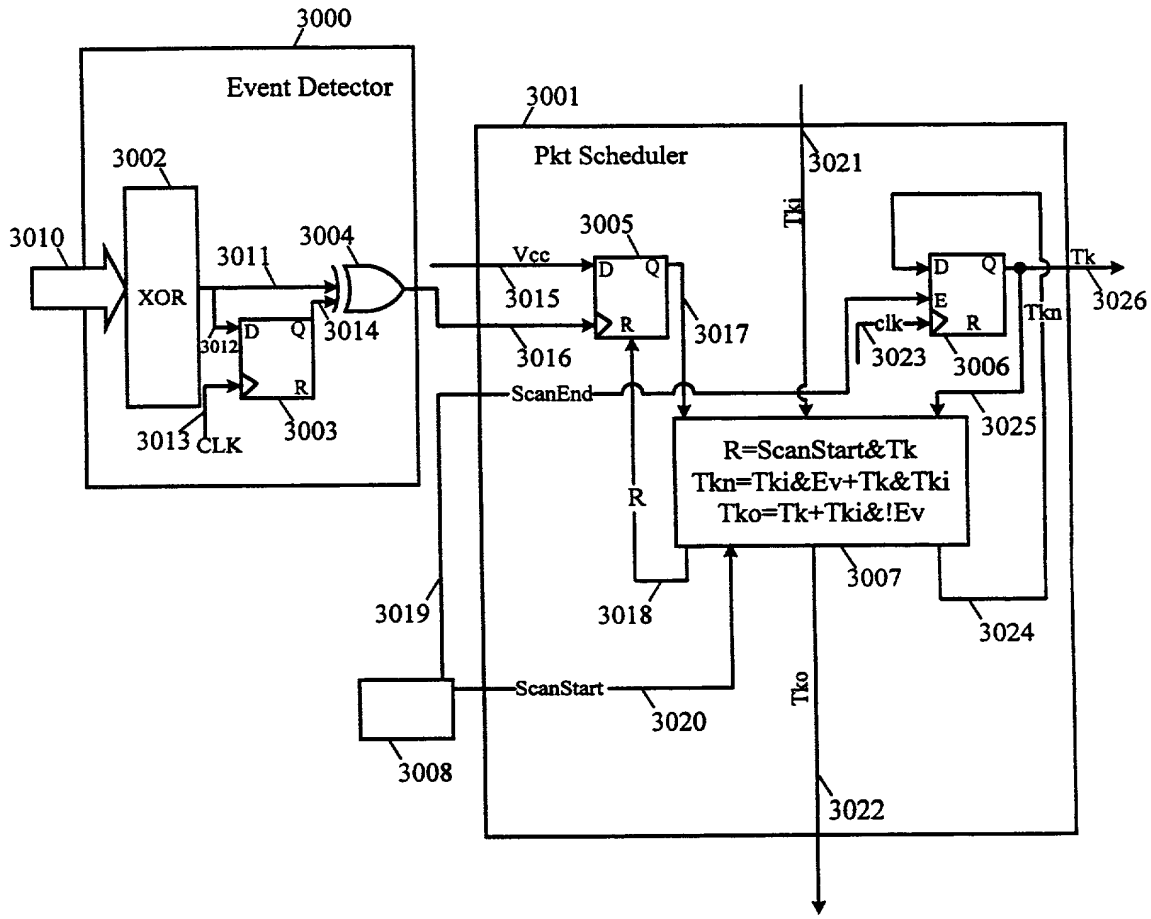

3000
Event Detector
3001
Pkt Scheduler
3002
3010
XOR
3011
3004
3012
D
Q
3014
R
3003
3013
CLK
3005
Vcc
3015
3016
3017
3021
Tki
clk
3023
3006
E
Tk
Tkn
3026
ScanEnd
3025
R=ScanStart&Tk
Tkn=Tki&Ev+Tk&Tki
Tko=Tk+Tki&!Ev
R
3019
3018
3007
3024
ScanStart
3020
3008
Tko
3022

FIG. 97

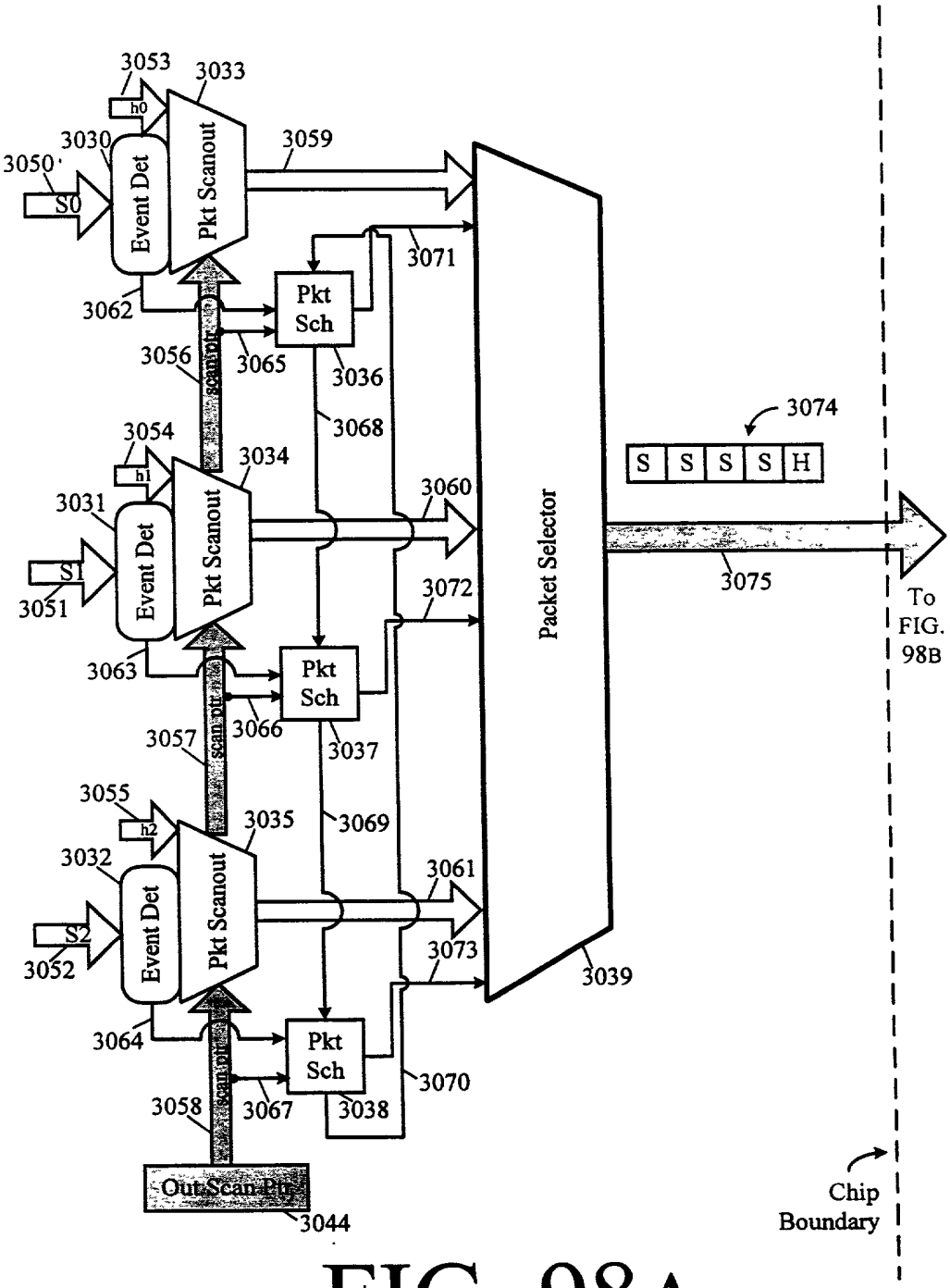

3053
3033
h0
3030
3050
S0
Event Det
Pkt Scanout
3059
3071
3062
Pkt Sch
3056
scan out
3065
3036
3068
3054
h1
3034
3031
3060
S1
3051
Event Det
Pkt Scanout
3072
3063
Pkt Sch
3057
scan out
3066
3037
3055
h2
3035
3069
3032
3061
S2
3052
Event Det
Pkt Scanout
3073
3039
3064
Pkt Sch
3070
3058
scan out
3067
3038
Out Scan Ptr
3044
Packet Selector
3074
S S S S H
3075
To FIG. 98B
Chip Boundary

FIG. 98A

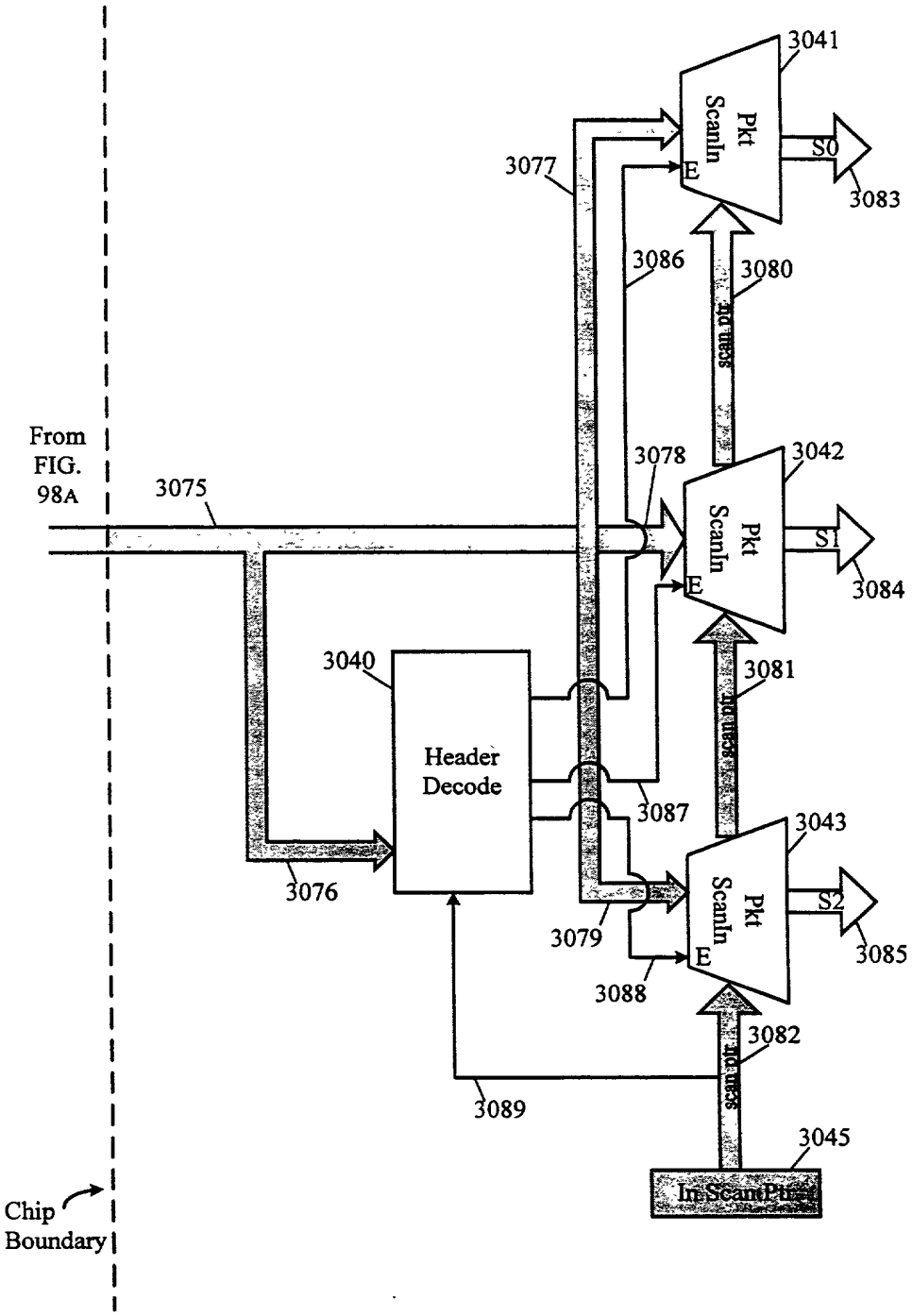

From FIG. 98A
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3040
Header Decode
3041
3042
3043
3045
Pkt ScanIn
E
S0
S1
S2
scan out
In ScanPtrs
Chip Boundary

FIG. 98B

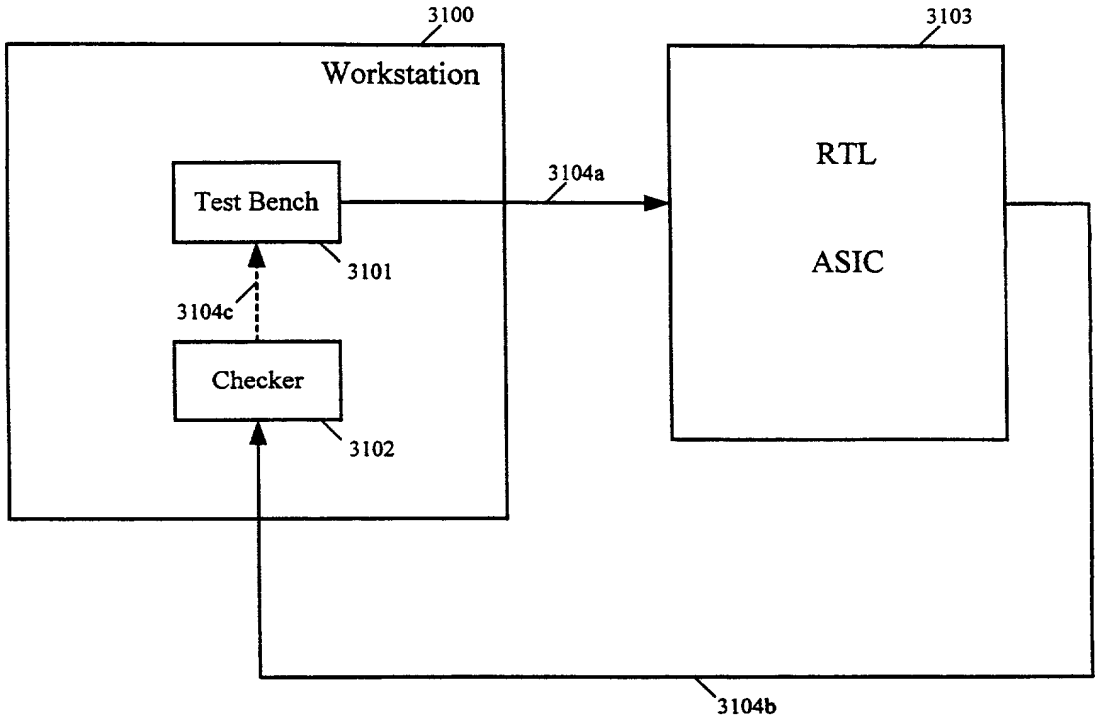

3100
Workstation
Test Bench
3101
3104c
Checker
3102
3104a
3103
RTL
ASIC
3104b

FIG. 99

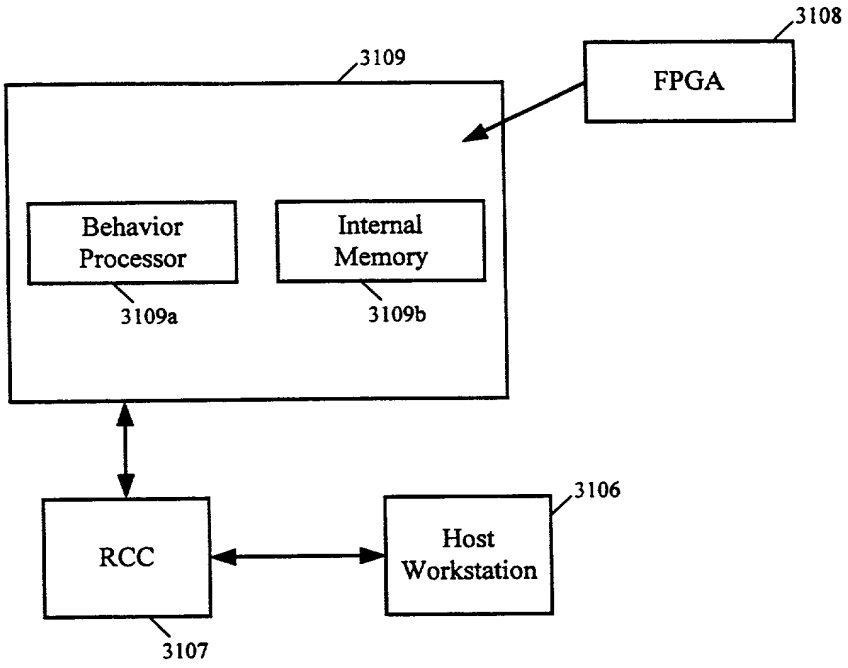

3108
FPGA
3109
Behavior Processor
3109a
Internal Memory
3109b
3106
Host Workstation
RCC
3107

FIG. 100

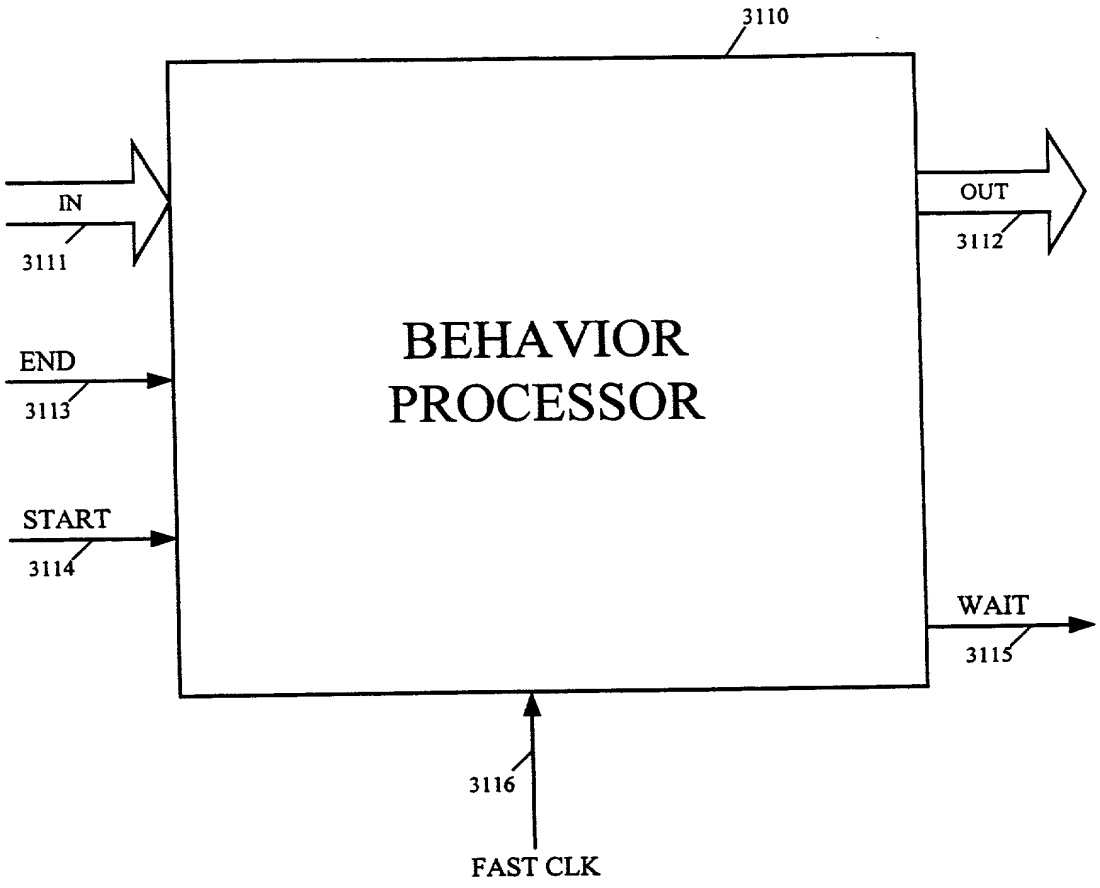

3110
IN
3111
END
3113
START
3114
BEHAVIOR
PROCESSOR
OUT
3112
WAIT
3115
3116
FAST CLK

FIG. 101

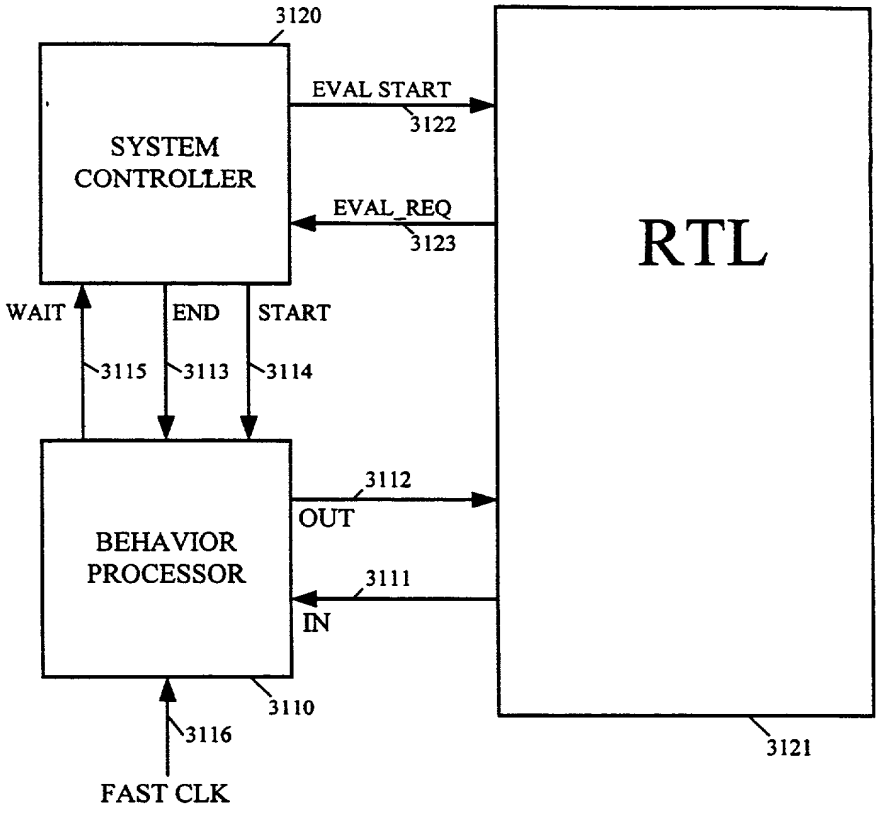

3120
SYSTEM CONTROLLER
EVAL START
3122
EVAL_REQ
3123
RTL
WAIT
END
START
3115
3113
3114
BEHAVIOR PROCESSOR
3112
OUT
3111
IN
3110
3116
FAST CLK
3121

FIG. 102

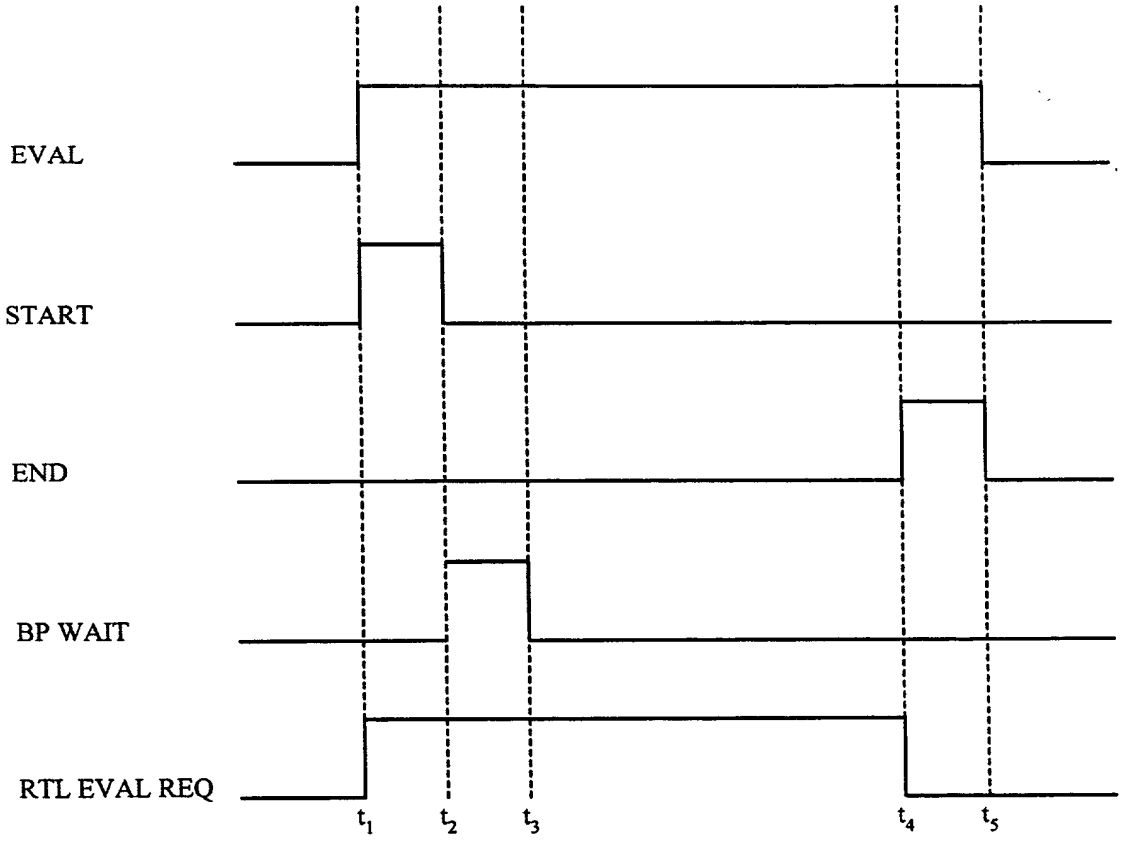


FIG. 103

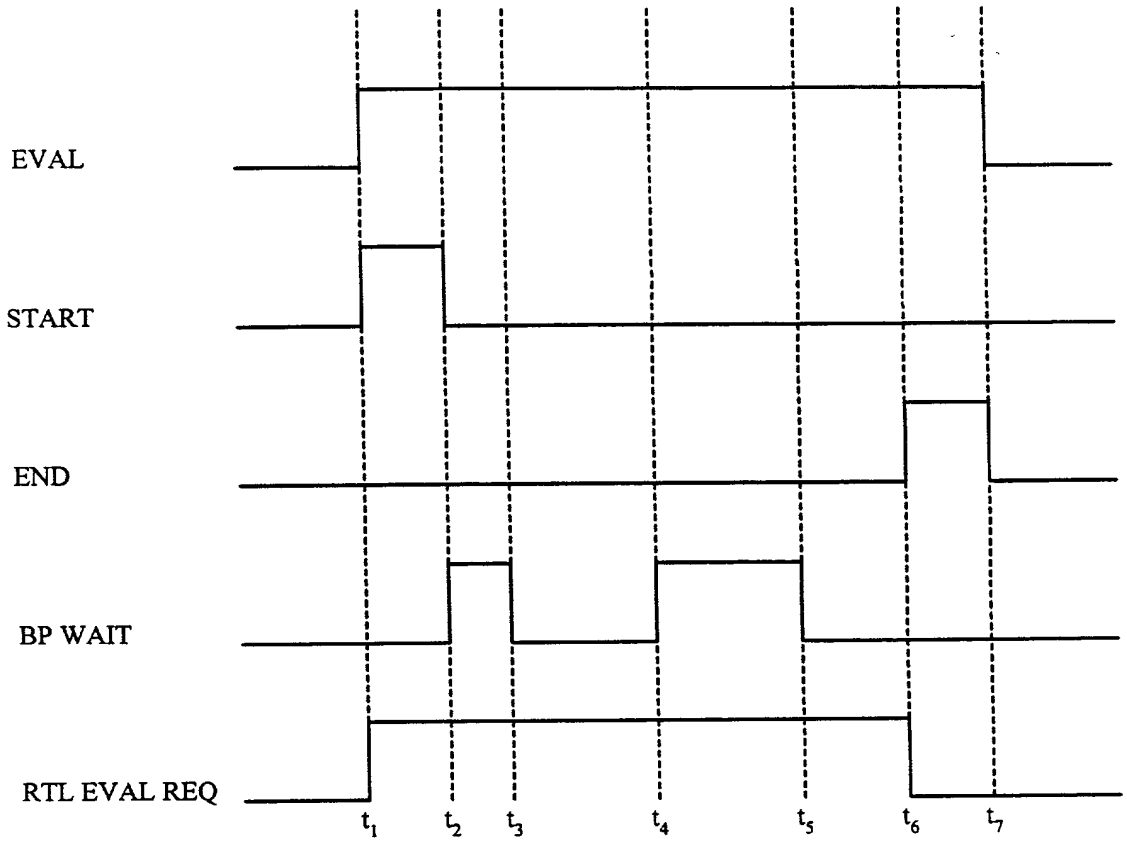


FIG. 104

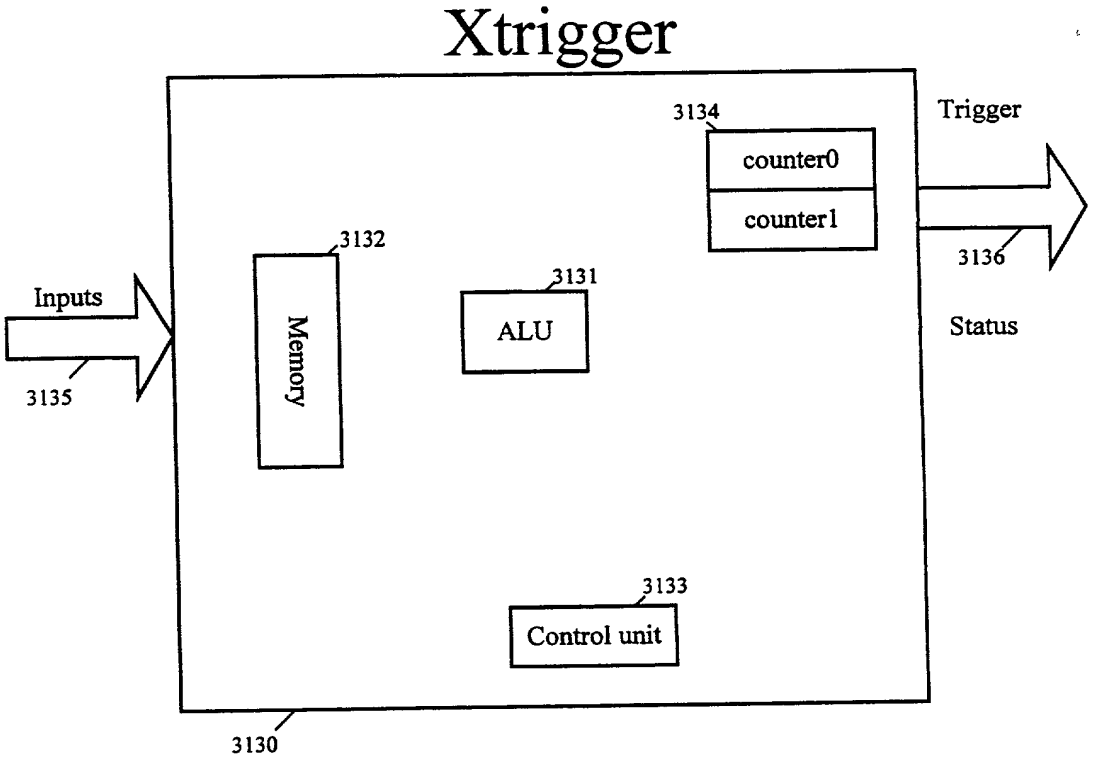

Xtrigger
3134
counter0
counter1
Trigger
3136
Status
3132
Memory
3131
ALU
Inputs
3135
3133
Control unit
3130

FIG. 105